第27卷 第3期　　北 京 工 业 大 学 学 报　　Vol.27 No.3
2001年 9月　　JOURNAL OF BEIJING POLYTECHNIC UNIVERSITY　　Sep. 2001

广义超立方网络的容错寻径算法研究

刘永峰[1]，刘长河[2]，寿玉亭[2]
（1. 北京建筑工程学院 机电系，北京 100044；2. 北京建筑工程学院 基础部，北京 100044）

摘　要：给定一个广义超立方网络 $G(m,r):N=m^r$ $(m\geqslant 2,\ r\geqslant 1)$，其上有若干条连线发生故障，$F$ 为其故障连线集合，且 $G(m,r)-F$ 是连通的，S 和 D 是 $G(m,r)$ 中任意两个结点(处理器)，其汉明距离 $H(S,D)=h$. 得出如下结论：(1) 当 $|F|<d$ 时，存在一条非故障路径 $P(S,D)$，且 $|P(S,D)|\leqslant h+2$；(2) 当 $d\leqslant|F|<m(d-m+1)$ 时，存在一条非故障路径 $P(S,D)$，且 $|P(S,D)|\leqslant h+4m-2$. 这里，d 是 $G(m,r)$ 的度，$|P(S,D)|$ 是路径 $P(S,D)$ 的长度，$P(S,D)$ 是非故障的是指在其上的所有连线均非故障. 给出了寻径算法.

关键词：广义超立方网络；容错；寻径算法
中图分类号：TP 301　　**文献标识码**：A　　**文章编号**：0254-0037(2001)03-0273-09

互连网络的容错寻径问题，是目前研究多处理器系统结构的热点之一[1~4]；它具有广泛的应用前景，尤其是在机械电子方面. 目前在此方面的研究很多. 但大多数工作都是针对 n 维超立方网络进行的，而对广义超立方 (GHC) 网络则研究较少. 文献 [5] 对 GHC 网络在某些处理器(结点)发生故障时的容错寻径算法进行了探讨；作者则在假设某些连线发生故障的情况下对同一问题进行研究，并给出相应的寻径算法. 关于 GHC 的结构与特性，请参阅文献 [6] 或 [7]，这里从略.

引理 1[5]　对于一个 r 维 GHC 网络上的任意两个两结点 A, B, $H(A,B)=h\leqslant r$, 存在 d 条互不相交的，长度 $\leqslant h+2$ 的路径 $P(A,B)$. 其中 $H(A,B)$ 为 A, B 之间的汉明距离，d 为各结点的度.

两条路径互不相交(或互相平行)，指除了源结点和目标结点外，在这两条路径上没有其他公共结点.

1　$|F|<d$ 时的情形

下面仅就 $\sqrt[r]{N}$ 为整数的情形进行讨论. 取 $m_i=\sqrt[r]{N}=m, 1\leqslant i\leqslant r$，即 $N=m^r$. 这时，$d=(m-1)r$.

模型：给定一个广义超立方网络 $G(m,r):N=m^r(m\geqslant 2, r\geqslant 1)$，其上有若干条连线发生故障；设其故障连线集合为 F，记 $|F|$ 为集合 F 中的元素个数. 用 (f_1,f_2) 表示结点 f_1 和 f_2 之间的连线.

由引理 1 知，此时至少有一条长度 $\leqslant h+2$ 的路径是畅通的.

本文的目的是要寻找从源结点 S 到目的结点 D 的寻径算法. 整个工作分为两部分：第一，寻找是否具有长度为 h 或 $h+1$ 的路径；第二，寻找是否具有长度为 $h+2$ 的路径. 为此定义一种新的“加法”，仍记为“$\oplus$”. 规定两个 r 位的地址数字相加，将其相同字位上的数字分别相比较，两者相等取 0，不等取 1. 比如：$(132)\oplus(012)=(110)$. 此时结点 (132) 和 (012) 间的距离即为 (110) 的质量 2. 显然，当 $m_i=2, 1\leqslant i\leqslant r$ 时，它和二进制中的模 2 加法是一致的.

记 $J(S,D)=\{j_1<j_2<\cdots<j_h\}$；$\alpha=S\oplus D=(0\cdots1\cdots1\cdots1\cdots1\cdots0)$；$\beta=f\oplus S=(0\cdots1\cdots1\cdots1\cdots0)$；
$\gamma=f\oplus D=(0\cdots1\cdots1\cdots1\cdots0)$；$\beta_1=f_1\oplus S=(0\cdots1\cdots1\cdots1\cdots0)$；$\beta_2=f_2\oplus S=(0\cdots1\cdots1\cdots1\cdots0)$；
$\gamma_1=f_1\oplus D=(0\cdots1\cdots1\cdots1\cdots0)$；$\gamma_2=f_2\oplus D=(0\cdots1\cdots1\cdots1\cdots0)$.

收稿日期：2001-01-16.
基金项目：北京建筑工程学院青年科研基金资助项目.
作者简介：刘永峰(1973-)，男，讲师，硕士.

这些符号中省略部分均为 0 或 1. 记 $G_h(S,D)$ 为 $G(m,r)$ 的仅由其第 $j_1<j_2<\cdots<j_h$ 维所构成的子广义超立方，它是包含 S 和 D 的最小的子广义超立方. 则:

(1) 如果 $\alpha\vee\beta_1=\alpha$ 且 $\alpha\vee\beta_2=\alpha$，说明 f_1 和 f_2 均在 $G_h(S,D)$ 中，连线 (f_1,f_2) 将阻断长度 $\leqslant h+1$ 的路径.

(1.1) 如果结点 f 的任何第 $j_i(1\leqslant i\leqslant h)$ 维地址数字，或与 S 相同，或与 D 相同，则 f 位于一条长度为 h 的路径上. 如果结点 f_1 和 f_2 均在同一条长度为 h 的路径上，则 (f_1,f_2) 阻断该条长度为 h 的路径.

(1.2) 如果结点 f 的某一第 $j_i(1\leqslant i\leqslant h)$ 维地址数字，既不与 S 相同，又不与 D 相同，则 f 位于一条长度为 $h+1$ 的路径上(注意: 恰在一维上不同，超过一维时，它不在长度为 $h+1$ 的路径上). 如果结点 f_1 和 f_2 均在同一条长度为 $h+1$ 的路径上，则 (f_1,f_2) 阻断该条长度为 $h+1$ 的路径.

(2) 如果 $\alpha\vee\beta_1\neq\alpha$ 或 $\alpha\vee\beta_2\neq\alpha$，说明 f_1 和 f_2 至少有一个结点不在 $G_h(S,D)$ 中，(f_1,f_2) 不阻断长度 $\leqslant h+1$ 的路径. 这时，如果结点 f_1 和 f_2 均在同一条长度为 $h+2$ 的路径上，则 (f_1,f_2) 阻断该条长度为 $h+2$ 的路径.

第一步，用 check(1.1) 来检查 (f_1,f_2) 是否阻断长度为 h 的路径. 如果 β 中含有 α 中连续的"1 块"，并且 $\beta\wedge\gamma_i=0$ (保证 f 的每位数字或与 S 相同, 或与 D 相同)，那么，f 一定在一条长度为 h 的路径上. 这可以如下检查: 首先计算数组 Index[1…r]. 如果字位 $i\notin J(S,D)$，置 Index[i] = 0; 如果 $i=j_k$，置 Index[i] = k. 对于每个 β，找其最右边的非 0 字位 r' 和最左边的非 0 字位 l'. 如果 weight(β) = Index[r'] − Index[l'] + 1，那么它是一个由连续的"1 块"组成的直接块，即无卷绕; 否则，在所有 $J(S,D)$ 字位对 β 取补，重复上述过程，即可检查 β 是否含有卷绕块. 如果它是一个块，那么 f 在以此块的开头字位开始的长度为 h 的路径上. 另外引进二元数组 Index(1.1)[1…r]. Index(1.1)[i] = 1 当且仅当以第 i 维开始的长度为 h 的路径被 (f_1,f_2) 所阻断，否则，Index(1.1)[i] = 0.

```
Procedure   check(1.1)(G(m,r), S, D, F)
  begin
    for all i = 1 to r do
      Index[i] = 0;
      Index(1.1)[i] = 0;
    end for
      α = S ⊕ S; c = 0;
    for i = 1 to r do    /* 建立数组 Index[1…r] */
      shift left α;
      if the bit shifted out is 1  then c = c + 1; Index[i] = c;  end if
    end for
    for all faulty links (f_1, f_2) ∈ F do
      for l = 1 to 2 do
      β_l = f_l ⊕ S; γ_l = f_l ⊕ D;
      π_l = β_l ∧ γ_l;
      if π_l = 0 then     /* f_1 的每位数字至少与 S 或 D 之一的相应字位相同 */
        b_l = block(1.1)(f_1);
      end if
    end for
    if (b_1 = A and b_2 ≠ A and b_2 ≠ null) then  Index(1.1)[b_2] = 1; end if
                                      /* 以 b_2 维开始的长度为 h 的路径被阻断 */
    if (b_2 = A and b_1 ≠ A and b_1 ≠ null) then  Index(1.1)[b_1]=1; end if
    if (b_1 = A and b_2 = A) then  Index(1.1)[1…r] = f_1 ⊕ f_2; end if  /*(f_1, f_2) = (S, D)*/
    if (b_1 = b_2 ≠ null and b_1 ≠ A) then  Index(1.1)[b_1]=1;  end if
```

```
  end for
end
function  block(1.1)(f);  /* 返回"null"或 A 或 f 所在的长度为 h 的路径的开始维数 */
begin
  β = f ⊕ S; γ = f ⊕ D;
  if α ∨ β = α  then   /*f 在包含 S 和 D 的最小子立方 G_h(S, D)中 */
    if (β = 0 or γ = 0)  then   /*f 与 S 或 D 重合 */
      return A;  /*f 在引理 1 中任何一条长度 ≤ h 的路径上 */
    else
    k_1 = frmb(β); k_2 = flmb(β);  /* 找β的最右(左)非 0 字位序号 */
    if  weight(β)=Index(k_1) − Index(k_2) + 1  then  return k_2;  /* 直接块 */
      else β′ = α ⊕ β  /* 对β的 J (S, D)字位取补 */
      k′_1 = frmb(β′); k′_2 = flmb(β′);
      if  weight(β′) = Index(k′_1)(Index(k_2′) + 1  then
        s = Index(k_1) + 1; k = j_s
        return  k;
      end if  /* 卷绕块 */
     end if
    end if
   else
     return  null;
  end if
end
```

第二步，检查连线(f_1,f_2)是否阻断长度为 $h+1$ 的路径．在 check(1.2)中，用到 mr 维向量 Index(1.2)[mr]． Index(1.2)[$i\cdot j$] = 1 当且仅当引理 1 证明中的 (2)中以$(i\cdot j)$为起始的长度为 $h+1$ 的路径被阻断，否则 Index(1.2)[$i\cdot j$] = 0． 程序 function block(1.2)(f) 用来判断f是否位于长度为 $h+1$ 的路径上．

```
Procedure  check(1.2)(G(m,r), S, D, F)
  begin
    for all i = 1 to  r  do
      for all j = 1 to m  do
        Index(1.2)[i · j] = 0
      end for
    ·nd for
      α = S ⊕ D;
    for all faulty links (f_1,f_2) ∈ F  do
      for l = 1 to 2 do
        β_l = f_l ⊕ S; γ_l = f_l ⊕ D;
      end for
      if (β_1 = 0 or γ_1 = 0) then  k = block(1.2)(f_2);  end if  /*f_1 与 S 或 D 重合 */
      if (β_2 = 0 of γ_2 = 0) then  k = block(1.2)(f_1);  end if  /*f_2 与 S 或 D 重合 */
      if ((β_1 and γ_1 and β_2 and γ_2) ≠ 0) then    /*f_1, f_2, S, D 互不重合 */
        k_1 = block(1.2)(f_1);  k_2 = block(1.2)(f_2);
        if k_1 = k_2 ≠ null  then  k = k_1;  end if
```

```
      end if
      Index(1.2)[k] = 1;
    end for
  end
  function  block(1.2)(f);  /* 返回"null"或f所在的长度为 h + 1 的路径的开始维数 */
  begin
    α = S ⊕ D; β = f ⊕ S; γ = f ⊕ D;
    if α ∨ β = α then  /*f 在包含 S, D 的最小广义超立方 G_h(S, D)中 */
      π = β ∧ γ; c = 0;
      for i = 1 to r do
        shift left π;
        if the bit shifted out is 1 then  c = c + 1; b = i;  end if;
      end for
      if c = 1 then  find the b th bit of f, f_b;  Return  b · f_b;
      end if      /*f 在以 [b · f_b]开始的长度为 h + 1 的路径上 */
     else
      return null;  /*f不在引理 1 证明中任何长度为 h + 1 的路径上 */
    end if
  end
```

第 3 步，检查连线(f_1,f_2)是否阻断长度为 $h+2$ 的路径．由引理 1 的证明过程可知，如果结点 (S, D 除外)f位于长度为 $h+2$ 的路径上须满足：

(1) 在 $J(S,D)$以外的字位上，f 恰有一位数字与 S 不同．

(2) 在 $J(S,D)$内的字位上，

(2.1) f的每位数字或与 S 相同，或与 D 相同．

(2.2) $S \oplus f$ 在 $J(S,D)$内的各位数字为$(1\cdots10\cdots0)$，即前面 k 个数字为 1，后面 $h-k$ 个数字为 0.

程序 function block2(f) 用来判断f是否位于长度为 $h+2$ 的路径上．

```
function  block2(f);  /* 返回"null"或f所在的长度为 h + 2 的路径的开始维数 */
begin
  S_1 = Mask all J(S, D) bits of  S;  /* 将 S 的 J(S, D) 维数字置 0*/
  f' = Mask all J(S, D) bits of  f;
  if weight(S_1 ⊕ f') = 1  then   /*f 和 S 在 J(S, D) 以外的字位上恰有一位数字不同. */
  α = S ⊕ D;  β = f ⊕ S;
  γ = f ⊕ D;  π = β ∧ γ;
  if π = 0  then /*f在 J(S, D) 的字位上的数字或与 S 相同，或与 D 相同. */
    S_2 = Mask all bits out of J(S, D) of S;
    f' = Mask all bits out of J(S, D) of  f;
    for all i = 1 to r do
    Index2[i] = 0
    end for
    σ = S_2 ⊕ f';  c = 0;
  for i = 1 to r do
    shift left σ
    If the bit shifted out is 1 then c = c + 1;  Index2[i]=c;  end if
  if (flmb(σ) = j_1 and weight(σ) = (frmb(σ) − (flmb(σ) + 1 ) /* 条件 (2.2)*/
```

```
      then  k = flmb(S_1 ⊕ f′);   find the kth bit of f, f_k;   return (k · f_k);
    end if
   end if
  else return null;
 end if
end
```

若故障连线(f_1,f_2)两端点有一个与S或D重合，另一个不在$G_h(S,D)$内，则(f_1,f_2)阻断长度为$h+2$的路径．若f_1、f_2均不在$G_h(S,D)$内，且f_1、f_2在同一条长度为$h+2$的路径上，则(f_1,f_2)阻断该路径.

```
 Procedure  check(2)(G(m, r), S, D, F)
 begin
   for all faulty links (f_1, f_2)  do
     F_0 = f_1;   F_1 = f_2;
     for l = 0 to 1 do
       β_l = F_l ⊕ S;  γ_l = F_l ⊕ D;
     if β_l = 0     /*F_l 与 S 重合 */
           Mask all J(S, D) bits of S;
           Mask all J(S, D) bits of F_{l⊕1};
           if weight(S ⊕ F_{l⊕1}) = 1 then    /* 连线 (S, F_{l⊕1}) ∉ J(S, D)*/
                 k(l)=flmb(S ⊕ F_{l⊕1});
                 find the k(l)th bit of F_{l⊕1}, F_{l⊕1,k(l)};
                 Index2[ k(l) · F_{l⊕1,k(l)} ] = 1;
           end if
     else if γ_l = 0      /*F_l与 D 重合 */
           Mask all J(S,D) bits of D;
           Mask all J(S,D) bits of F_{l⊕1},
           if weight(D ⊕ F_{l⊕1}) = 1 then    /* 连线 (D, F_{l⊕1}) ∉ J(S,D)*/
             k(l) = flmb(D ⊕ F_{l⊕1});
             find the k(l)th bit of F_{l⊕1}, F_{l⊕1,k(l)};
             Index2 [ k(l) · F_{l⊕1,k(l)} ] = 1;
           end if
     else    /*β_l ≠ 0 and γ_l ≠ 0, F_l 不与 S, D 重合 */
         Mask all J(S,D) bits of S;
         Mask all J(S,D) bits of F_l;
         Mask all J(S,D) bits of F_{l⊕1};
         if weight (S ⊕ F_l)+ weight (S ⊕ F_{l⊕1}) = 2  then  /*F_0, F_1 均在 G_h(S,D)外 */
           k_l = block2 (F_l);   k_{l⊕1} = block2 (F_{l⊕1});
           if k_l = k_{l⊕1} ≠ null  then  Index2[k_l]=1;  end if
         end if
       end if
     end for
 end for
end
```

最后，用下面程序完成$|F|<d$时的寻径.

```
Procedure  route1(G(m, r), S, D, F)
```

```
begin
  check (1.1)(G(m, r), S, D, F).  /* 检查是否有长度为 h 的路径 */
  for i = 1 to r  do
    if  i ∈ J(S, D)  and  Index(1.1)[i] = 0  then
      return the minimal path starting from i and exit;
    end if
  end for
  check (1.2)(G(m, r), S, D, F)  /* 检查是否有长度为 h + 1 的路径 */
  for i = 1 to r  do
   for j = 1 to m − 1  do
     if  i ∈ J(S, D)  and  Index(1.2) [i · j] = 0  then
       return the path with length of h + 1 starting form (i · j) and exit;
     end if
   end for
  end for
  check2 (G(m, r), S, D, F)  /* 检查是否有长度为 h + 2 的路径 */
  for i = 1 to r  do
   for j = 1 to m − 1  do
    if  i ∉ J(S,D) and Index (2) [i · j] = 0  then
       return the path with length of h + 2 starting form (i · j) and exit;
    end if
   end for
  end for
nd
```

定理1 对于一个 r 维广义超立方网络 $G(m,r)$: $N = m^r$ 其度为 d. 如果其中故障连线的个数 $|F| < d$, 则对于 $G(m,r)$ 中任意两个结点 S, D, 一定可以找到一条非故障路径 $P(S,D)$: $|P(S,D)| \leqslant H(S,D) + 2$. 当 $H(S,D) = r - 1$ 且 S 在 $G(m,r-1)$ 内是孤立的时, 其最小路径的长度为 $r + 1$. 所以当 $|F| < d$ 时, $G(n\ r) - F$ 的直径 $\leqslant r + 1$ (因为当 $H(S,D) = r$ 时一定存在一条长度 $\leqslant r + 1$ 的非故障路径 $P(S,D)$).

2 $|F| \geqslant d$ 的情形

如果故障连线的数目 $|F| \geqslant d$, 将不能保证具有长度 $\leqslant H(S,D) + 2$ 的非故障路径 $P(S,D)$. 下面考察当 $d \leqslant |F| < m(d - m + 1)$ 时的情形.

给定一个广义超立方网络 $G(m,r)$: $N = m^r$, 则其每个结点的度 $d(m,r) = (m - 1)r$[5,6], 沿着某一维 $k($ $\leqslant k \leqslant r)$, 可将 $G(m,r)$ 分割成 m 个 $r - 1$ 维的 GHC 网络 $G(m,r-1)$: $G_0(m,r-1)$, $G_1(m,r-1)$, $\cdots$, $G_{\ \ 1}(m,r-1)$. S 和 D 是其中任意两个结点, 现在寻找从 S 到 D 的非故障路径.

2. 特殊情形: 当有"好维"存在时

一个维 i 被称为"好维", 如果和 S, D 连接的所有第 i 维连线均非故障.

定理2 给定一个 r 维广义超立方网络 $G(m,r)$: $N = m^r$, 其度为 d, 其中故障连线的个数 $|F| < m(\ - m + 1)$. S, D 是 $G(m,r)$ 中任意两个结点, 且在包含 S, D 的最小广义超立方网络 $G_h(S,D)$ 中有"好维"存在, 则一定存在一条长度至多为 $H(S,D) + 4$ 的非故障路径.

对于 $m = 2$ (超立方网络) 的情形, 文献 [5] 已有较全面的论述, 这里假设 $m \geqslant 3$.

证明 假设 $J(S,D) = \{j_1 < j_2 < \cdots < j_h\}$. 设有一"好维" i, 沿着 i, S 的邻接结点为 X_i, D 的邻接结点

为 $Y_l, l = 0, 1, \cdots, m-1$(将 S 看成$\{X_l\}$中的一个特殊结点，将 D 看成$\{Y_l\}$中的一个特殊结点)．

(1) 若 $i \in J(S, D)$

沿着 i 将 G_r 分成 m 个G_{r-1}^l; $l = 0, 1, \cdots, m-1$(每个 G_{r-1}^l的度为 $d-m+$)．则划分之后，这 m 个 G_{r-1}^l中，必有一个(不妨记为G_{r-1}^0)包含 S 与某 $Y_l(\neq D$, 不妨记为 $Y_0)$，另一个(不妨记为G_{r-1}^1)包含 D 与某 $X_l(\neq S$, 不妨记为 $X_1)$，而其他$m-2$ 个 G_{r-1}中必包含一个 $X_l(\neq S)$ 和一个 $Y_l(\neq D)$．由引理 1 易知，从 S 到 D 存在 $2(d-m+1)$ 条长度 $\leqslant (h+2+1=)h+3$ 的路径和 $(m-2)(d-m+1)$ 条长度 $\leqslant (h+2+2=)h+4$ 的路径．于是共有 $m(d-m+1)$ 条长度 $\leqslant h+4$ 的路径，其中至少有一条不被故障连线所阻断．

(2) 若 $i \notin J(S, D)$，从而 $h < r$

沿着 i 将 G_r 分成 m 个G_{r-1}^l; $l = 0, 1, \cdots, m-1$．则 S 和 D 必在某 G_{r-1}^l(不妨记为G_{r-1}^0)中，而其他$m-1$ 个 G_{r-1}中必包含一个 $X_l(\neq S)$ 和一个 $Y_l(\neq D)$．不妨假设 G_{r-1}^l中包含 X_l 和 Y_l; $l = 1, \cdots, m-1$．由引理 1 易知，从 S 到 D 存在 $(d-m+1)$ 条长度 $\leqslant h+2$ 的路径和 $(m-1)(d-m+1)$ 条长度 $\leqslant (h+2+2=)h+4$ 的路径．于是共有 $m(d-m+1)$ 条长度 $\leqslant h+4$ 的路径，其中至少有一条不被故障连线所阻断．

下面给出此时的寻径算法．先利用 access 找“好维”：

```
Procedure access
begin
  a = b = 0;
  for all faulty links (f1,f2)  do
    δ = f1 ⊕ S;  γ = f2 ⊕ S;
    if weight(δ)=weight(γ) = 1 then a = a ∨ δ ∨ γ; end if
    δ = f1 ⊕ D;  γ = f2 ⊕ D;
    if weight(δ) = weight(γ) = 1 then b = b ∨ δ ∨ γ; end if
  end for
  c = ā ∧ b̄;
end
```

向量 c 中数字 1 所在的维均是“好维”．如果具有“好维”存在，可用 route2 来实现此时的寻径问题．

```
Procedure route2(G(m,r), S, D, F)
 begin
   if c ∧ (S ⊕ D) ≠ 0  then
     pick a good dimension i ∈ J(S, D);
     P(S, D) = route1(G^0_{r-1}, S, Y_0, F∩ G^0_{r-1}) ∘ (i · k_0);  /*(i · k_0)将 Y_0的第 i 位数字变为与 D 相同 */
     if P(S, D) = null  then P(S, D) = (i · k_1)route1(G^1_{r-1}, X_1, D, F∩ G^1_{r-1});
     end if               /*(i · k_1)将 S 的第 i 位数字变为与 X_1相同 */
     if P(S, D) = null  then
       for l = 2 to m − 1 do
         P(S, D) = (i · l) ∘ Route1(G^l_{r-1}, X_l, Y_l, F∩ G^l_{r-1}) ∘ (i · l); /* 前一个(i · l)将 S 的第 i 位数
                                    字变为与 X_l 相同，后一个(i · l)将 Y_l 的第 i 位
                                    数字变为与 D 相同．下面一个与此相同．*/
       end for
     end if
   else        /* 没有 i ∈ J(S, D)的好维存在 */
     pick any good dimension i;
     P(S, D) = route1(G^0_{r-1}, S, D, F∩ G^0_{r-1});
     if P(S, D) = null  then
       for l = 1 to m − 1 do
```

```
      P(S, D) = (i · l) ∘ Route1(G'_{r-1}, X_p, Y_p, FI G'_{r-1}) ∘ (i · l);  /* (i · l)同上 */
      if P(S, D) ≠ null then exit; end if
    end for
  end if
 end if
end
```

2.2 一般情形

引理 2 沿任一维 k 将 $G(m,r)$ 分割成 m 个 $r-1$ 维子 GHC 网络：$G_i(m,r-1)$；$i=0,1,\cdots,m-1$. 则对于 $G_0(m,r-1)$ 中的任一结点 A，可以分别找到 $d=(m-1)r$ 条长度 $\leqslant 2$ 的、互不相交的路径分别通向每个 $G_i(m,r-1)$ $(i=1,2,\cdots,m-1)$ 中的 d 个不同的结点.

证明 只证明 $A\to G_1(m,r-1)$ 的情形. $A\to G_i(m,r-1)$ $(i=2,3,\cdots,m-1)$ 的情形同理可证.

假设 $A=a_1\cdots a_{k-1}0a_{k+1}\cdots a_r\in G_0(m,r-1)$ 则,

$$P_k: A\to a_1\cdots a_{k-1}1a_{k+1}\cdots a_r\in G_1(m,r-1) \tag{4}$$

是一条长度为 1 的路径.

$$P_j: A\to a_1\cdots\bar{a}_j\cdots a_{k-1}0a_{k+1}\cdots a_r(\in G_0(m,r-1))\to a_1\cdots\bar{a}_j\cdots a_{k-1}1a_{k+1}\cdots a_r\in G_1(m,r-1) \tag{5}$$

其中：$\bar{a}_j\in\{0,1,\cdots,m-1\}-\{a_j\}$，$j\in\{1,2,\cdots,r\}-\{k\}$. 这种路径共 $(m-1)(r-1)$ 条，其长度均为 2.

$$P_k': A\to a_1\cdots a_{k-1}la_{k+1}\cdots a_r(\in G_l(m,r-1))\to a_1\cdots a_{k-1}1a_{k+1}\cdots a_n\in G_l(m,r-1) \tag{6}$$

其中 $l\in\{0,1,\cdots,m-1\}-\{0,1\}$. P_k' 的长度也为 2，和 P_j 不同的是，它绕道 $G_l(m,r-1)$，再进入 $G_1(m,r-1)$ 中，这种路径有 $(m-2)$ 条. 显然，这一组的 $(m-1)r$ 条路径是互不相交的.

任取 $k_i\in J(S,D)$，沿第 k_i 维可将 $G(m,r)$ 分成 m 个 $r-1$ 维子 GHC：$G_l(m,r-1)$；$l=0,1,\cdots,m-1$. S 和 D 不在同一个子立方中. 不妨设 $S\in G_0(m,r-1)$，$D\in G_1(m,r-1)$，则此时至少有一个子 GHC（不妨设为 $G_L(m,r-1)$）中故障连线的个数 $<|F|/m\leqslant(m-1)(r-1)=d(m,r-1)$. 寻找引理 2 中的路径 $P_{L_i}: S\to S_L\in G_L(m,r-1)$，$Q_{L_i}: D\to D_L\in G_L(m,r-1)$. 这时，$S_L$、$D_L$ 同在 $G_L(m,r-1)$ 中. 注意，若 $L=0$，则 $S_L=S$；若 $L=1$，则 $D_L=D$. 由引理 1，在 $G_L(m,r-1)$ 中存在一条长度 $\leqslant H(S_L,D_L)+2$ 的非故障路径 $P(S_L,D_L)$. 利用算法 Procedure route1 $(G_L(m,r-1),S_L,D_L,F,G_L(m,r-1))$ 找出 $P(S_L,D_L)$ 则：

$P(S,D)=S\xrightarrow{P_{L_i}}S_L\xrightarrow{P(S_L,D_L)}D_L\xrightarrow{Q_L^{-1}}D$. 其中路径 $Q_{L_i}^{-1}$ 表示 Q_{L_i} 的逆向路径.

接下来寻找路径 $P_{L_i}: S\to S_L\in G_L(m,r-1)$：

① 从 S 出发到 $G_L(m,r-1)$ 中的路径有 d 条，如果这些路径中有一条是非故障的，取该路径为 P_{L_i}.

② 如果①中所有路径全部被阻断，由于 $G(m,r-1)-F$ 的连通性，可沿非故障连线找到 S 的一个 u_1，从 u_1 到 $G_L(m,r-1)$ 中也有 d 条长度 $\leqslant 2$ 的路径，并且这一组路径中只有 m 条与①中的路径有公共结点，因此，从 S 到 $G_L(m,r-1)$ 中的互不相交的路径增加了 $d-m$ 条. 若这些新增加的路径中有一条是非故障的，则由该路径：$S\to u_1\xrightarrow{P_{l_i}}S_L\in G_L(m,r-1)$ 可找到 S_L；否则，寻找 u_1 的邻接结点，记为 u_2，且 $d(S,u_2)=2$. 从 u_2 出发又有 d 条通向 $G_L(m,r-1)$ 的长度 $\leqslant 2$ 的路径，且只和前面两组路径中的 m 条有公共结点… 重复以上过程，可一直找到 u_{m-1}，使得，$d(S,u_{m-1})=m-1$，$d(u_1,u_{m-1})=m-2$，…，$d(u_{m-2},u_{m-1})=1$. 这时不相交的路径的总数 $\geqslant md-(m-1)m=m(d-m+1)$. 而 $|F|<m(d-m+1)$，从而，一定有一条路径 P_{L_i} 非故障. 这样，沿路径：$S\to u_1\to\cdots\to u_{m-1}\xrightarrow{P_{L_i}}S_L\in G_L(m,r-1)$ 进入 $G_L(m,r-1)$，这时，$H(S_L,D)\leqslant H(S,D)+(m-1)$.

③ 同样，可找到一条从 D 到 $G_L(m,r-1)$ 中的结点 D_L 的非故障路径.

从上面的分析可知：$H(S_L,D_L)\leqslant H(S,D)+2(m-1)$. 由定理 1，在 $G_L(m,r-1)$ 中可找到一条路径 $P(S_L,D_L)$，且 $|P(S_L,D_L)|\leqslant H(S_L,D_L)+2\leqslant H(S,D)+2m$. 于是，$|P(S,D)|\leqslant|P(S_L,D_L)|+2(m-1)\leqslant H(S,D)+2(2m-1)$.

定理 3 如果广义超立方网络 $G(m,r)$：$N=m^r$ 中的故障连线集合 F 满足：$d\leqslant|F|<m(d-m+1)$，且 $G(m,r)-F$ 是连通的，则对于 $G(m,r)$ 中的任意两个结点 S,D，一定有一条非故障路径 $P(S,D)$：$|P$

$(S,D)| \leqslant h+2(2m-1)$. 这里，$d$ 是 $G(m,r)$ 的度，$h=H(S,D)$ 为 S 和 D 之间的汉明距离.

本文的算法对于超立方网络也适用.

参考文献:

[1]ESFAHANIAN A H. Generalized measures of fault tolerance with application to N-cube networks. IEEE Trans Comput, 1989, 38(11): 1586-1591.

[2]LALIFI S. Combinatorial analysis of the fault diameter of the n-cube. IEEE Trans Comput, 1993, 42(1): 27-33.

[3]TIEN Sing-Ban, RAGHAVENDRA C S. Algorithms and bounds for shortest paths and diameter in faulty hypercubes[J]. IEEE Trans Parallel and Distributed Systems, 1993, 4(6): 713-718.

[4]QIAN Ping, PENG Shietung. An efficient algorithm for node-to-node routing in hypercubes with faulty clusters[J]. The Computer Journal, 1996, 39(1): 14-19.

[5]童明生，刘长河，范天佑. 一般化超立方网络的容错寻径算法. 计算机学报，1998，21(12)：1074-1083.

[6]王鼎兴，陈国良. 互连网络结构分析. 北京：科学出版社，1990.

[7]BHUYAN L, AGRAWAL D P. Generalized hypercube structures for a computer network[J]. IEEE Trans Comput, 1984, C-33(44): 323-333.

Research on Fault-Tolerance Routing Algorithms on Generalized Hypercube Networks

LIU Yong-feng[1], LIU Chang-he[2], SHOU Yu-ting[2]

(1. Department of Mechanical and Electrical Engineering, Beijing Institute of Civil Engineering and Architecture, Beijing 100044, China;

2. Department of Basic Science, Beijing Institute of Civil Engineering and Architecture, Beijing 100044, China)

Abstract: The fault-tolerance routing algorithms on generalized hypercube networks with the set of faulty links are studied. Let $G(m,r)$: $N=m^r$ $(m \geqslant 2, r \geqslant 1)$, be a generalized hypercube networks with the set of faulty links F, $|F|$ denotes the number of elements in F and the graph $G(m,r)-F$ is connected. S and D are two nodes, between which the Hamming distance is $H(S,D)=h$. Then, there are conclusions as following: (1) If $|F|<d$, there is a fault-free path $P(S,D)$, such that $|P(S,D)| \leqslant h+2$; (2) If $d \leqslant |F| < m(d-m+1)$, there is a fault-free path $P(S,D)$, such that $|P(S,D)| \leqslant h+4m-2$. where, $|P(S,D)|$ is the length of Path $P(S,D)$ and d is the degree of $G(m,r)$. Path $P(S,D)$ is fault-free means that there is no faulty links on it. The corresponding routing algorithms are proposed in this paper.

Key words: generalized hypercube networks; fault-tolerance; routing algorithm