

# Torus 网络中容错路由算法的设计与概率分析

李 银, 梁家荣, 徐 霜, 肖 杰

LI Yin, LIANG Jia-rong, XU Shuang, XIAO Jie

广西大学 计算机与电子信息学院, 南宁 530004

College of Computer, Electronics and Information, Guangxi University, Nanning 530004, China

E-mail: yinli020608@126.com

LI Yin, LIANG Jia-rong, XU Shuang, et al. Design of fault tolerant routing algorithm and probabilistic analysis on Torus networks. *Computer Engineering and Applications*, 2009, 45(14): 103-106.

**Abstract:** In the paper, a simple fault-tolerant routing algorithm is presented based on the concept of  $k$ -subtorus. Under the assumption that each node has an independent failure probability, it is able to compute the probability of the fault-free routing path which found by the routing algorithm. For the Torus network which has more than hundreds of thousands nodes, it is at least 99% the probability that a fault-free routing path can be found by the routing algorithm presented in the paper, which runs in linear time.

**Key words:** Torus networks; fault tolerant routing algorithm; connectivity; probabilistic analysis

**摘 要:** 基于  $k$ -Torus 子网的概念提出了一个简单的 Torus 网络容错路由算法。假设结点出错相互独立, 计算出路由算法成功路由的概率。对于几十万个结点以上的 Torus 网络, 提出的路由算法构造通路的概率可达 99%, 且所提出的路由算法具有线性的特点。

**关键词:** Torus 网络; 容错路由算法; 连通性; 概率分析

**DOI:** 10.3778/j.issn.1002-8331.2009.14.031 **文章编号:** 1002-8331(2009)14-0103-04 **文献标识码:** A **中图分类号:** TP302

## 1 引言

直连网络(Direct Interconnection Network, DIN)<sup>[1]</sup>是一种常见的网络拓朴形式, 已经广泛应用于多处理器系统(Multi-processor), 多计算机系统(Multi-computer), 以及集群系统(cluster)中。网格(Mesh)网络是人们较早研究的一种直连网络<sup>[2]</sup>, 它结构规则, 简单易于实现, 但是 Mesh 结构不对称, 会极大地影响网络性能<sup>[3]</sup>。Torus 网络则是一种完全对称的直连网络拓朴形式<sup>[4]</sup>。Torus 网络具有很多优秀的网络特性, 如规则对称性, 路径多样性以及良好的扩展性, 因此广泛应用于许多商用系统中, 例如 2004 年底评出的全球超级计算机 TOP100 中排名首位的 IBM BlueGene/L 就采用 Torus 网络<sup>[5]</sup>; 而另一家通信设备制造商, Avici 公司在其推出的世界上第一台太比特路由器中也采用 Torus 网络作为其交换网络拓朴<sup>[6]</sup>。可见 Torus 网络已经成为最为重要的网络模型之一。

随着计算机网络规模的不断扩大, 网络容错性成为一个不可回避的研究课题。如何在结点出错或链路出错的情况下保证多处理器系统能够连续正常的运行? 容错路由算法就是为解决此问题而设计的。目前关于 Torus 网络容错性的研究成果并不多见, 鉴于 Torus 网络在实际应用中的重要性, 本文的目的就是通过分析  $k$ -Torus 子网提出一个可靠的容错路由算法, 并给

出其容错性的概率分析, 为 Torus 网络的应用提供有力的理论基础。

本文的结构如下, 第 2 章给出了一些相关的概念和定义; 第 3 章提出了一个基于  $k$ -Torus 子网的简单的基于局部信息和分布式的容错路由算法, 并基于结点出错概率研究了路由算法的容错性, 计算出不同规模的 Torus 网络在要求算法成功返回一条由正确结点组成的路径的概率为不小于 99% 时对结点出错概率的要求; 最后在第 4 章中给出了结论。

## 2 相关概念和定义

**定义 1**<sup>[4]</sup> 用  $T_{k_1 \times k_2 \times \dots \times k_{n-1}}$  表示  $n$  维 Torus 网络, 是由  $k_0 \times k_1 \times \dots \times k_{n-1}$  个结点构成, 其中  $k_i$  表示第  $i$  维的结点数。用一个  $n$  维向量  $(x_0, x_1, \dots, x_{n-1})$  来标识网络上的每一个结点, 其中  $0 \leq x_i \leq k_i - 1$ 。结点  $(x_0, x_1, \dots, x_{n-1})$  和结点  $(y_0, y_1, \dots, y_{n-1})$  相连接的条件是 iff  $\exists i$  使  $x_i = (y_i \pm 1) \bmod k_i$ , 而  $\forall j \neq i$ , 有  $x_j = y_j$ 。

研究规模为  $m \times n$  的二维 Torus 网络, 该方法很容易推广到任意维的 Torus 网络中去。类似于文献[5]中采用的子网划分方法, 给出  $k$ -Torus 子网的定义。

**定义 2** 给定 Torus 网络  $T_{m \times n}$ , 一个  $k$ -Torus 子网用符号  $T_{b,d}^{(k)}$

**基金项目:** 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60564001); 国家教育部新世纪人才支持计划(the New Century Excellent Talent Foundation from MOE of China under Grant No.NCET-06-0756); 广西研究生教育创新计划资助项目(No.2007105930812M49)。

**作者简介:** 李银(1984-), 女, 硕士, 主要研究方向: 计算机网络、路由算法、容错性; 梁家荣(1966-), 男, 博士, 教授, 主要研究领域: 数据挖掘、模糊信息处理等; 徐霜(1982-), 女, 硕士, 主要研究方向: 并行计算; 肖杰(1984-), 男, 硕士, 主要研究方向: 网络容错。

**收稿日期:** 2008-03-12 **修回日期:** 2008-05-15

表示,其中  $b < mlk, d < nlk$ , 相应的组成子网节点的坐标可以表示为  $(x, y)$ , 其中  $x = bk, bk+1, \dots, (b+1)k-1, y = dk, dk+1, \dots, (d+1)k-1$  (为了讨论简单, 假设  $m$  和  $n$  为  $k$  的整数倍, 当不为整数倍时同样适用但结果会有微小的差异)。

按照定义 2, 图 1 给出了  $6 \times 6$  的 Torus 网络被分成 4 个的 3-Torus 子网的情形。

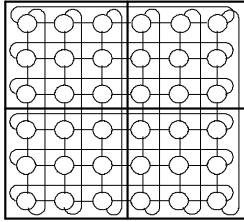


图 1  $k$ -Torus 子网划分策略

从图 1 可以看出, 一个规模为  $m \times n$  的 Torus 网络  $T_{m \times n}$  可以分成  $(m/k) \times (n/k)$  个相互独立的  $k$ -Torus 子网, 每一个  $k$ -Torus 子网有 4 个边界, 为方便下文的叙述, 沿用文献[5]中相关的符号表示方法:

(1)  $T_k$  为一个  $k$ -Torus 子网, 四周边分别用  $L$  (左边界)、 $R$  (右边界)、 $U$  (上边界) 及  $D$  (下边界) 表示。

(2) 事件  $A_X(T_k)$ :  $X$  为集合  $\{L, R, U, D\}$  的子集, 事件  $A_X(T_k)$  要求  $T_k$  中  $X$  边界上正确结点数大于错误结点数, 且正确结点之间存在通路。

(3)  $p$ : 结点具有独立的出错概率  $p$ ;  $\Pr[A_X(T_k)]$ : 事件  $A_X(T_k)$  的概率;  $a_{k,i}^{(X)}$ :  $X$  为集合  $\{L, R, U, D\}$  的子集,  $a_{k,i}^{(X)}$  表示从  $T_k$  中去掉  $i$  个结点且满足事件  $A_X(T_k)$  要求的方法个数。

(4) 事件  $D(T_{m \times n}^{A \rightarrow B})$ : 算法成功返回一条从结点  $A$  到  $B$  的正确结点组成的路径。  $\Pr[D(T_{m \times n}^{A \rightarrow B})]$ : 事件  $D(T_{m \times n}^{A \rightarrow B})$  的概率。

有如下引理。

引理 1 事件  $A_{LR}(T_k)$  和事件  $A_{UD}(T_k)$  的概率为

$$\Pr[A_{LR}(T_k)] = \Pr[A_{UD}(T_k)] = \sum_{i=0}^{k^2} a_{k,i}^{(LR)} (1-p)^{k^2-i} p^i$$

事件  $A_{LU}(T_k), A_{LD}(T_k), A_{RU}(T_k), A_{RD}(T_k)$  的概率为

$$\Pr[A_{LU}(T_k)] = \Pr[A_{LD}(T_k)] = \Pr[A_{RU}(T_k)] = \Pr[A_{RD}(T_k)] = \sum_{i=0}^{k^2} a_{k,i}^{(LU)} (1-p)^{k^2-i} p^i$$

事件  $A_L(T_k), A_R(T_k), A_U(T_k), A_D(T_k)$  的概率为

$$\Pr[A_L(T_k)] = \Pr[A_R(T_k)] = \Pr[A_U(T_k)] = \Pr[A_D(T_k)] = \sum_{i=0}^{k^2} a_{k,i}^{(L)} (1-p)^{k^2-i} p^i$$

事件  $A_{LRUD}(T_k)$  的概率为

$$\Pr[A_{LRUD}(T_k)] = \sum_{i=0}^{k^2} a_{k,i}^{(LRUD)} (1-p)^{k^2-i} p^i$$

证明 其证明方法类似于文献[5]。

### 3 容错 Torus 网络路由算法设计及其概率分析

#### 3.1 容错路由算法设计

假设需要构造一条从结点  $A$  到结点  $B$  的路径。

设  $A$  属于  $k$ -Torus 子网  $T_{u,v}^{(k)}$ , 而  $B$  属于  $k$ -Torus 子网  $T_{u',v'}^{(k)}$ ,

不失一般性, 假设  $u \leq u', v \leq v'$ 。由于 Torus 网络的拓扑结构增加了环回信道<sup>[6]</sup>, 为了避免如图 2 所示的维内死锁, 引入虚信道技术来避免图中所示死锁的产生。这里规定, 将每条物理信道分为两条虚信道, 分别标识为 0 和 1, 路由时首选 0 信道, 当遇到边缘信道后用 1 信道进行路由, 因此不会出现图 2 所示的环路从而达到死锁避免。

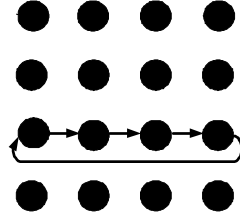


图 2 Torus 中可能出现的环路

#### 算法 1

输入: 具有故障结点的二维 Torus 网络  $T_{m \times n}$  及两个正确结点  $A(x_a, y_a)$  和  $B(x_b, y_b)$ 。

输出:  $T_{m \times n}$  中由结点  $A$  到结点  $B$  的一条正确路由  $P$ 。

步骤 1 根据  $u = \lfloor x_a/k \rfloor, v = \lfloor y_a/k \rfloor, u' = \lfloor x_b/k \rfloor, v' = \lfloor y_b/k \rfloor$  确定结点  $A$  所在的  $k$ -Torus 子网  $T_{u,v}^{(k)}$  及结点  $B$  所在的  $k$ -Torus 子网  $T_{u',v'}^{(k)}$ ; // 由于 Torus 网络具有对称性, 不妨设  $u \leq u', v \leq v'$ 。

步骤 2  $d_x = \lfloor n/2k \rfloor, d_y = \lfloor m/2k \rfloor$ ; //  $(d_x, d_y)$  为划分子网后的中间子网的标号。

步骤 3 初始化路径及当前结点  $C: P \leftarrow \{A\}, C \leftarrow A$ 。

步骤 4 IF  $((u \geq d_x) \parallel (u < d_x \ \&\& \ u' \leq d_x + u))$  // 直接向右扩展路径且用 0 信道

FOR  $j = u$  TO  $u' - 1$  DO

① 在子网  $T_{j,v}^{(k)}$  的  $R$  边界与子网  $T_{j+1,v}^{(k)}$  的  $L$  边界寻找一对连通的结点  $w'$  和  $w''$ ;

② 扩展路径  $P$ : 将结点  $C$  到结点  $w'$  和  $w''$  的路径加入  $P$  中,  $C \leftarrow w''$ 。

ELSE IF  $(u < d_x \ \&\& \ u' > d_x + u)$  // 向左扩展, 利用绕边路由

(1) FOR  $j = 1$  TO  $u$  DO // 用 0 信道路由;

① 在子网  $T_{u-j+1,v}^{(k)}$  的  $L$  边界与子网  $T_{u-j,v}^{(k)}$  的  $R$  边界寻找两个相邻正确结点  $w'$  和  $w''$ ;

② 扩展路径  $P$ : 将结点  $C$  到结点  $w'$  和  $w''$  的路径加入  $P$  中,  $C \leftarrow w''$ 。

(2) 在子网  $T_{0,v}^{(k)}$  的  $L$  边界和子网  $T_{n/k-1,v}^{(k)}$  的  $R$  边界寻找两个相邻正确结点  $w'$  和  $w''$ ; 将路径  $P$  在  $T_{0,v}^{(k)}$  中从  $C$  扩展到  $w'$ , 然后扩展到  $w''$ ,  $C \leftarrow w''$ ;

// 边界子网绕道路由, 用 1 信道路由。

(3) FOR  $i = 1$  TO  $n/k - u' - 1$  DO // 用 0 信道路由;

① 在子网  $T_{n/k-i,v}^{(k)}$  的  $L$  边界和子网  $T_{n/k-i-1,v}^{(k)}$  的  $R$  边界寻找两个相邻正确结点  $w'$  和  $w''$ ;

② 扩展路径  $P$ : 将结点  $C$  到结点  $w'$  和  $w''$  的路径加入  $P$  中,  $C \leftarrow w''$ 。

步骤 5 IF  $((v \geq d_y) \parallel (v < d_y \ \&\& \ v' \leq d_y + v))$  // 直接向上扩展路径且用 0 信道

FOR  $k = v$  TO  $v' - 1$  DO

① 在子网  $T_{u',k}^{(k)}$  的  $U$  边界和子网  $T_{u',k+1}^{(k)}$  的  $D$  边界寻找两个

相邻正确结点  $w'$  和  $w''$ ;

②扩展路径  $P$ : 将结点  $C$  到结点  $w'$  和  $w''$  的路径加入  $P$  中,  $C \leftarrow w''$ 。

ELSE IF  $(v < d_y \ \&\& \ v' > d_y + v) //$  向下扩展, 利用绕边路由

(1) FOR  $k=1$  TO  $v$  DO // 用 0 信道路由;

①在子网  $T_{u', v-k+1}^{(k)}$  的  $D$  边界和子网  $T_{u', v-k}^{(k)}$  的  $U$  边界寻找两个相邻正确结点  $w'$  和  $w''$ ;

②扩展路径  $P$ : 将结点  $C$  到结点  $w'$  和  $w''$  的路径加入  $P$  中,  $C \leftarrow w''$ 。

(2) 在子网  $T_{u', 0}^{(k)}$  的  $D$  边界和子网  $T_{u', mlk-1}^{(k)}$  的  $U$  边界寻找两个相邻正确结点  $w'$  和  $w''$ ; 将路径  $P$  在  $T_{u', 0}^{(k)}$  中从  $C$  扩展到  $w'$ , 然后扩展到  $w''$ ,  $C \leftarrow w''$ ;

// 边界子网绕道路由, 用 1 信道路由。

(3) FOR  $h=1$  TO  $mlk-v-1$  DO // 用 0 信道路由

①在子网  $T_{u', mlk-h}^{(k)}$  的  $D$  边界和子网  $T_{u', mlk-h-1}^{(k)}$  的  $U$  边界寻找两个相邻正确结点  $w'$  和  $w''$ ;

②扩展路径  $P$ : 将结点  $C$  到结点  $w'$  和  $w''$  的路径加入  $P$  中,  $C \leftarrow w''$ 。

步骤 6 在  $k$ -Torus 子网  $T_{u', v'}^{(k)}$  中寻找一条从  $C$  到达结点  $B$  的路径;

步骤 7 算法终止, 找到一条从结点  $A$  到结点  $B$  的正确路由  $P$ ;

// 如果算法在以上任何一步无法执行, 则算法终止, 未能找到一条从结点  $A$  到结点  $B$  的正确路由  $P$ 。

接下来分析算法的时间复杂性。根据 Torus 网的特点, 容易看出本算法无论在  $X$  方向上还是  $Y$  方向上路由, 所经过的子网个数都不大于  $\lfloor k_i/2 \rfloor$ ,  $k_i$  为  $X$  或  $Y$  方向上的子网个数。因此, 在最坏的情况下, 算法需要在  $\lfloor n/2k \rfloor + \lfloor m/2k \rfloor + 1$  个  $k$ -Torus 子网中扩展路径, 即步骤 4 最多为  $\lfloor n/2k \rfloor$  个子网, 步骤 5 最多为  $\lfloor m/2k \rfloor$  个子网, 步骤 6 则为子网  $T_{u', v}^{(k)}$ 。在子网中扩展路径可以采用 BFS 或 DFS 搜索, 无论采用哪种方法, 两者的时间复杂度相同。因为每一  $k$ -Torus 子网有  $k^2$  个结点和  $2k^2$  条边, 因此在每个子网进行路径扩展可在时间  $O(k^2)$  中完成。于是, 算法 1 的时间复杂性为  $O((\lfloor n/2k \rfloor + \lfloor m/2k \rfloor)k^2) = O((\lfloor n/2 \rfloor + \lfloor m/2 \rfloor)k)$ , 当  $k$  值取得很小时, 算法 1 在最坏情况下的时间复杂性为线性的。

### 3.2 算法 1 容错性的概率分析

定理 1 算法 1 在最坏情况下以至少  $(\Pr[A_L(T_k)])^2 \cdot (\Pr[A_{LR}(T_k)])^{\lfloor n/2k \rfloor + \lfloor m/2k \rfloor - 2} \cdot (\Pr[A_{LU}(T_k)])$  的概率构造一条从结点  $A$  到结点  $B$  的正确结点组成的路径。

证明 对  $T_{u', v}^{(k)}$  的  $R$  或  $L$  边界上的任一正确结点  $w$ , 如果满足事件  $A_R(T_k)$  或  $A_L(T_k)$  的要求, 就可以在  $T_{u', v}^{(k)}$  中从结点  $A$  路由到结点  $w$ 。根据事件  $A_{LR}(T_k)$  与  $A_{LD}(T_k)$  的定义, 知道算法路径上任何相邻子网的相邻边界上正确结点数大于错误结点数, 因此必定能在相邻边界上找到至少一对相邻正确结点, 使相邻子网连通, 因此能够构造一条从源结点到目的结点的无错路径。具体分析如下:

(1) 在  $X$  轴上, 如果直走, 那么当事件  $A_R(T_{u', v}^{(k)}), A_{LR}(T_{u'+1, v}^{(k)}), \dots, A_{LR}(T_{u'-1, v}^{(k)}), A_L(T_{u', v}^{(k)})$  同时发生, 那么可以路由到与目的结

点所在子网具有相同  $X$  轴标号的子网上。

(2) 如果绕边路由, 那么当事件  $A_L(T_{u', v}^{(k)}), A_{LR}(T_{u'-1, v}^{(k)}), \dots, A_{LR}(T_{0, v}^{(k)}), A_{LR}(T_{mlk-1, v}^{(k)}), \dots, A_{LR}(T_{u'+1, v}^{(k)}), A_R(T_{u', v}^{(k)})$  同时发生, 那么可以路由到与目的结点所在子网具有相同  $X$  轴标号的子网上。

同理在  $Y$  轴上, 可以路由到与目的结点所在子网具有相同  $Y$  轴标号的子网上。

(3) 如果直走, 那么当事件  $A_U(T_{u', v}^{(k)}), A_{UD}(T_{u', v+1}^{(k)}), \dots, A_{UD}(T_{u', v-1}^{(k)}), A_D(T_{u', v}^{(k)})$  同时发生, 那么可以路由到与目的结点所在子网具有相同  $Y$  轴标号的子网上。

(4) 如果绕边路由, 那么当事件  $A_D(T_{u', v}^{(k)}), A_{UD}(T_{u', v-1}^{(k)}), \dots, A_{UD}(T_{u', 0}^{(k)}), A_{UD}(T_{u', mlk-1}^{(k)}), \dots, A_{UD}(T_{u', v+1}^{(k)}), A_U(T_{u', v}^{(k)})$  同时发生, 那么可以路由到与目的结点所在子网具有相同  $Y$  轴标号的子网上。

成功构造一条从结点  $A$  到  $B$  的正确结点组成的路径即为路由到与目的结点所在子网具有相同  $X$  轴和  $Y$  轴标号的子网上, 那么事件  $D(T_{m \times n}^{A \rightarrow B})$  发生的概率至少为以上 4 种情况组合的情形之一, 即 (1)(3), (2)(3), (1)(4), (2)(4) 四种组合之一。由对称性可知

$$\Pr[A_R(T_k)] = \Pr[A_L(T_k)] = \Pr[A_U(T_k)] = \Pr[A_D(T_k)]$$

$$\Pr[A_{LR}(T_k)] = \Pr[A_{LD}(T_k)]$$

又由于算法在每维上所经过的子网个数都不大于  $\lfloor k_i/2 \rfloor$ ,  $k_i$  为  $X$  或  $Y$  方向上的子网个数, 因此

$$\begin{aligned} \Pr[D(T_{m \times n}^{A \rightarrow B})] &\geq (\Pr[A_L(T_k)])^2 \cdot (\Pr[A_{LR}(T_k)])^{\lfloor n/2k \rfloor - 1} \cdot \\ &(\Pr[A_{LU}(T_k)]) \cdot (\Pr[A_{UD}(T_k)])^{\lfloor m/2k \rfloor - 1} \geq \\ &(\Pr[A_L(T_k)])^2 \cdot (\Pr[A_{LR}(T_k)])^{\lfloor n/2k \rfloor + \lfloor m/2k \rfloor - 2} \cdot \\ &(\Pr[A_{LU}(T_k)]) \end{aligned}$$

定理 1 即得证。

下面, 结合定理 1 和算法 1 举例说明。令  $k=5$ , 首先计算事件  $A_L(T_5)$ 、事件  $A_{LR}(T_5)$  和事件  $A_{LU}(T_5)$  的概率。为了方便定理 1 的验证, 需要引用如下结论。

引理 2<sup>[5]</sup> 假设每个结点具有独立的出错概率  $p$ , 则事件  $A_{LR}(T_5)$  的概率为

$$\begin{aligned} \Pr[A_{LR}(T_5)] &= (1-p)^{25} + 25p(1-p)^{24} + 296p^2(1-p)^{23} + \\ &2\ 168p^3(1-p)^{22} + 10\ 809p^4(1-p)^{21} + \\ &37\ 820p^5(1-p)^{20} + 92\ 332p^6(1-p)^{19} + \\ &154\ 148p^7(1-p)^{18} + 174\ 629p^8(1-p)^{17} + \\ &138\ 152p^9(1-p)^{16} + 81\ 467p^{10}(1-p)^{15} + \\ &38\ 021p^{11}(1-p)^{14} + 14\ 482p^{12}(1-p)^{13} + \\ &4\ 577p^{13}(1-p)^{12} + 1\ 161p^{14}(1-p)^{11} + \\ &208p^{15}(1-p)^{10} + 19p^{16}(1-p)^9 \end{aligned}$$

引理 3<sup>[5]</sup> 假设每个结点具有独立的出错概率  $p$ , 则

$$\begin{aligned} \Pr[A_L(T_5)] &= (1-p)^{25} + 25p(1-p)^{24} + 296p^2(1-p)^{23} + \\ &2\ 178p^3(1-p)^{22} + 11\ 002p^4(1-p)^{21} + \\ &39\ 554p^5(1-p)^{20} + 101\ 706p^6(1-p)^{19} + \\ &186\ 932p^7(1-p)^{18} + 249\ 440p^8(1-p)^{17} + \\ &251\ 122p^9(1-p)^{16} + 200\ 589p^{10}(1-p)^{15} + \end{aligned}$$

$$\begin{aligned}
& 132\ 717p^{11}(1-p)^{14} + 75\ 134p^{12}(1-p)^{13} + \\
& 37\ 328p^{13}(1-p)^{12} + 16\ 610p^{14}(1-p)^{11} + \\
& 6\ 721p^{15}(1-p)^{10} + 2\ 503p^{16}(1-p)^9 + \\
& 871p^{17}(1-p)^8 + 291p^{18}(1-p)^7 + 95p^{19}(1-p)^6 + \\
& 31p^{20}(1-p)^5 + 11p^{21}(1-p)^4 + 3p^{22}(1-p)^3 + \\
\Pr[A_{LU}(T_5)] = & (1-p)^{25} + 25p(1-p)^{24} + 296p^2(1-p)^{23} + \\
& 2\ 168p^3(1-p)^{22} + 10\ 809p^4(1-p)^{21} + \\
& 37\ 856p^5(1-p)^{20} + 92\ 795p^6(1-p)^{19} + \\
& 156\ 800p^7(1-p)^{18} + 183\ 109p^8(1-p)^{17} + \\
& 153\ 569p^9(1-p)^{16} + 98\ 832p^{10}(1-p)^{15} + \\
& 51\ 692p^{11}(1-p)^{14} + 22\ 890p^{12}(1-p)^{13} + \\
& 8\ 895p^{13}(1-p)^{12} + 3\ 118p^{14}(1-p)^{11} + \\
& 1\ 002p^{15}(1-p)^{10} + 297p^{16}(1-p)^9 + 81p^{17}(1-p)^8 + \\
& 21p^{18}(1-p)^7 + 5p^{19}(1-p)^6 + p^{20}(1-p)^5
\end{aligned}$$

将这些值带入定理 1, 可以算出算法 1 在 Torus 网络中路由成功的概率。

此外, 对于制造商来说, 更为关心的是所构造的网络中处理器结点出错概率应保证在多大范围内, 而使网络能正常运行, 因此可以令  $(\Pr[A_L(T_k)])^2 \cdot (\Pr[A_{LR}(T_k)])^{\lfloor n/2k \rfloor + \lfloor m/2k \rfloor - 2} \cdot (\Pr[A_{LU}(T_k)]) \geq q$ , 从而求出结点出错概率  $p$  的一个上界。为此取  $q=99\%$ , 所得结果在表 1 中列出。

从表 1 可以得到, 只要结点出错概率控制在 0.48% 之内, 路由算法 1 能够保证正确结点之间存在通路。且由前面的算法时间复杂度的讨论知道, 此算法能够在线性时间内构造此通路。因此, 算法 1 是非常有效的。

(上接 62 页)

对于小规模实例, 本文的混合遗传算法在求解质量是很优秀的。但是对于稍微大一点的规模就不是那么好, 比如 KroA150, 本算法的最优结果就没达到 TSPLIB 中的最优解。

## 5 结论

在一般混合遗传算法的启发下, 提出了一种新的混合遗传算法。引入逆转算子(爬山法), 加快了算法的局部搜索能力。并根据 TSP 适应度地貌特征, 设计一种贪婪的复合变异算子, 提高遗传算法的全局搜索能力。多组实验测试表明, 特别是对小于 150 个城市的 TSP 实例测试, 效果比 TSPLIB 中的最优路径还要好。

## 参考文献:

- [1] 陈国良, 王煦天, 庄镇泉, 等. 遗传算法理论及其应用[M]. 北京: 人民邮电出版社, 1996.
- [2] 王小平, 曹立明. 遗传算法—理论应用与软件实现[M]. 西安: 西安交通大学出版社, 2002.
- [3] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1996.
- [4] 玄光男, 程润伟. 遗传算法与工程设计[M]. 北京: 科学出版社, 2000.
- [5] Merz P. A comparison of memetic recombination operators for the traveling salesman problem[C]//Proceedings of the Genetic and Evo-

表 1 当算法 1 成功构造一条由正确结点组成的路径的概率为 99% 时对网络结点出错概率  $p$  的要求

Torus 网络的大小	结点出错概率/(%)
50×50	1.41
100×100	1.04
200×200	0.75
400×400	0.54
500×500	0.48

## 4 结论

主要研究了 Torus 网络的容错路由算法, 并在结点出错独立的假设条件下对算法进行概率分析, 给出算法成功构造路径的概率公式。此外, 根据公式计算了结点出错概率的上界, 只要其概率控制在 0.48% 之内, 那么算法 1 就可以在两个正确结点之间构造通路且其成功的概率为 99%。

## 参考文献:

- [1] Dally W, Towles B. Principles and practices of interconnection networks[M]. San Francisco: Morgan-Kaufmann Press, 2004.
- [2] Liu Gu-qing, Chen Zhen, Qiu Zhi-liang, et al. Research on the link traffic and delay of a direct networks[J]. Journal of Xidian University, 2003, 30(7): 96-100.
- [3] 顾华玺, 刘增基, 王琨, 等. Torus 网络中分布式自适应路由算法[J]. 西安电子科技大学学报: 自然科学版, 2006, 33(3): 352-358.
- [4] Adiga N R, Almasi G S. An overview of the BlueGene/L supercomputer[C]//Proc of Super Computing, Baltimore: IEEE, 2002: 1-22.
- [5] 王高才, 陈建二, 陈松乔, 等. Mesh 网络路由算法容错性的概率分析[J]. 计算机学报, 2004, 27(3): 319-327.
- [6] 顾华玺, 邱智亮, 涂小行, 等. E-2D Torus 网络结构中的无死锁路由算法[J]. 小型微型计算机系统, 2005, 26(7): 1140-1144.
- [7] Evolutionary Computation Conference(GECCO), 2002: 472-479.
- [8] Baraglia R, Hidalgo J I, Perego R. A hybrid heuristic for the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 2001, 5: 613-622.
- [9] 邹鹏, 周智, 陈国良, 等. 求解 TSP 问题的多级归约算法[J]. 软件学报, 2003, 14(1): 35-42.
- [10] Michalewicz Z, Fogel D B. 如何求解问题—现代启发式方法[M]. 曹宏庆, 译. 北京: 中国水利水电出版社, 2003: 30-33.
- [11] Jones T, Forrest S. Fitness distance correlation as a measure of problem difficulty for genetic algorithms[C]//Eshelman L J. Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufman, San Francisco, CA, USA, 1995: 184-192.
- [12] 杨平, 郑金华. 遗传选择算子的比较和研究[J]. 计算机工程与应用, 2007, 43(15): 59-65.
- [13] Tao G, Michalewicz Z. Inver-over operator for the TSP[C]//LNCS: Proceedings of the 5 Parallel Problem Solving form Nature, 1998, 1498: 803-812.
- [14] Katayama K, Sakamoto H. The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem[J]. Mathematical and Computer Modelling, 2000, 31: 197-203.
- [15] 刘习春, 喻寿益. 局部快速微调遗传算法[J]. 计算机学报, 2006, 29(1): 100-105.
- [16] TSPLIB[EB/OL]. (2007-05-22). Http://www.lwr.uni\_heidelberg.de.