

文章编号: 2095—0411 (2011) 04—0050—06

# 容迟网络中的节点转发策略<sup>\*</sup>

史 超

(南京邮电大学 通信与信息工程学院, 江苏 南京 210003)

**摘要:** 容迟网络结构为很多面临挑战的网络提供了互操作性, 它没有稳定的端到端的传输路径, 而且具有高延迟和频繁中断的特点, 当前的 Internet 体系结构和网络协议无法适用于这样的网络。基因算法 (GA) 是启发式求解法, 建立在生物器官的遗传过程的基础上。并行基因进化论 (PGE) 是一种新的有效探索某问题解决空间的方法。在并行基因进化论 (PGE) 中, 群体被分为若干亚群体, 基因算法在每个亚群体中独立运作。主要研究在容迟网络中利用分布式基因进化理论, 研究基于并行基因进化论 (PGE) 的节点策略模型, 构建以质粒迁徙为基础的区域性基因算法, 研究高度分布式环境下的节点转发策略。

**关键词:** 容迟网络; 并行基因算法; 质粒迁徙; 节点转发策略

**中图分类号:** TP 915.9

**文献标识码:** A

## Research on Delay—Tolerant—Network of Natural Cooperation Technology

SHI Chao

(School of Telecommunication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** Delay—tolerant network provides interoperability for networks that are confronted with many challenges. Delay—tolerant network does not have any end—to—end transmission routes and is characterized by high delay and frequent interruption, so the present Internet architecture and network protocol can not apply to this kind of network. Genetic Algorithm (GA) is a heuristic approach based on genetic process of biological organs. Parallel Genetic Evolution (PGE) is a new and effective way to explore the solving space of a certain problem. In Parallel Genetic Evolution (PGE), populations are divided into sub—populations, in each of which Genetic Algorithm is practiced independently. Mainly studying the application of distributed genetic evolution theory in delay—tolerant network and node strategy model based on PGE, this paper builds a regional genetic algorithm on the basis of plasmid migration and studies node forwarding strategy in highly distributed circumstances.

**Key words:** delay—tolerant network; parallel genetic evolution (PGE); plasmid migration; node forwarding strategy

## 1 容迟网络和基因算法

容迟网络 (DTN) 是指能在长延时、连续频

繁断开等受限网络条件下进行通信的一种新型的网络体系<sup>[1]</sup>。这种网络结构为很多面临挑战的网络 (比如星际网络、传感器网络、陆地移动网络等)

<sup>\*</sup> 收稿日期: 2011—09—03

作者简介: 史超 (1988—), 男, 江苏连云港人, 本科。

提供了互操作性。容迟网络没有稳定的端到端的传输路径,而且具有高延迟和频繁中断的特点,当前的 Internet 体系结构和网络协议无法适用于这样的网络<sup>[2,3]</sup>。

基因算法最早由美国密执安大学的 Holland 教授提出,起源于 20 世纪 60 年代对自然和人工自适应系统的研究。70 年代 De Jong 基于基因算法的思想在计算机上进行了大量的纯数值函数优化计算实验。在一系列研究工作的基础上,80 年代由 Goldberg 进行归纳总结形成了基因算法的基本框架<sup>[4]</sup>。基因演算法是启发式求解法,建立在生物器官的基因遗传过程的基础上。

本文介绍容迟网络的概念及其相关发展状况,了解基因算法以及并行基因进化理论(PGE),在容迟网络中利用分布式基因进化理论,研究基于 PGE 的节点策略模型,构建以质粒迁徙为基础的区域性基因算法,研究高度分布式环境下的节点转发策略。

## 2 容迟网络中的基因算法

DTN 与 Internet 体系结构有着明显的差别。为了解决受限网络存在的问题,DTN 在 Internet 体系结构的基础上增加了一个束层(bundle layer),位于传输层之上。它将一些消息聚集在一起(称为束 bundle)进行转发,不同于 Internet 网关的分组交换<sup>[5]</sup>。DTN 体系结构在数据传输效率低、延迟过大、故障频发以及间断链接的情况下提供了可靠的通信方法。DTN 通过束层实现端到端的可靠传输。传送的数据会先被存放在发送端束层得缓冲区中,直到发送端确认数据已成功传送到接收端时,才将缓冲区中的数据删除,否则就重传一次。DTN 的路由在束层进行,而在 Internet 中路由只到网络层,因此在 DTN 中延迟主要发生在束层<sup>[6]</sup>。

典型的基因演算法作用于一群个体,其中每一个个体都代表一种某问题的解决方法,个体适合度的分值取决于它这一解决方法对于问题的适合程度,适合程度越高,被繁殖的可能性越高。在这一繁殖中,两个个体结合产生拥有上一代最优特征的下一代个体。随着一代代延续,这些最优的特征在真个群体中传播。借鉴这个最适合个体配对的过程,可能探求到空间里最前景的区域<sup>[7]</sup>。通过将基因算法在容迟网络中的合理应用,减少束层的延迟。

### 2.1 基因算法的运算过程

生物的进化是以集团为主体的。与此相对应,基因算法的运算对象是由  $M$  个个体所组成的集合,称为群体。与生物一代一代的自然进化过程相类似,基因算法的运算过程也是一个反复迭代过程,第  $t$  代群体记作  $P$  (经过一代遗传和进化后,得到第  $t+1$  代群体,它们也是由多个个体组成的集合,记作  $P(t+1)$ 。这个群体不断地经过遗传和进化操作,并且每次都按照优胜劣汰的规则将适应度较高的个体更多地遗传到下一代,这样最终在群体中将会得到一个优良的个体  $X$ ,它所对应的表现型  $X$  将达到或接近于问题的最优解  $X$ 。

生物的进化过程主要是通过染色体之间的交叉和染色体的变异来完成的。与此相对应,基因算法中最优解的搜索过程也模仿生物的这个进化过程,使用所谓的遗传算子(genetic operators)作用于群体  $P(t)$  中,进行下述遗传操作,从而得到新一代群体  $P(t+1)$ 。常见 3 种基因算法。①选择(selection):根据各个个体的适应度,按照一定的规则或方法,从第  $t$  代群体  $P(t)$  中选择出一些优良的个体遗传到下一代群体  $P(t+1)$  中;②交叉(crossover):将群体  $P(t)$  内的各个个体随机搭配成对,对每个对个体,以某个概率(称为交叉概率, crossover rate)交换他们之间的部分染色体;③变异(mutation):对群体  $P(t)$  中的每一个个体,以某一概率(称为变异概率, mutation rate)改变某一个或某一些基因座上的基因值为其他的等位基因。

### 2.2 平行基因演算法

平行基因演算法(PGA)不仅是典型基因演算法(GA)的延伸,还是一种新的有效探索某问题的解决空间的方法。在平行基因算法(PGA)中,群体被分为若干亚群体,基因演算法(GA)在每个亚群体中独立运作。区域选择和繁殖原则使得物种在区域内进化发展,而迁移,即亚群体间基因信息的转移,使得多样性增强。

平行基因演算法模式可以依据其平行级别分为岛或单元。在平行基因演算法模式岛(IPGA)中,整个群体被分成几个独立的包含若干个体的亚群体或岛,每个岛有一个独立的基因演算式运作。经过几代后,发生迁徙,其间最适个体在岛间迁移以保证生物多样性。在一个平行基因演算单元(cP-

GA) 中, 整个群体被分成大量独立的包含几个个体亚群体, 基因信息交流通过亚群体重叠进行。在平行基因演算单元 (cPGA) 中的群体有一种空间结构, 可以把个体交流限制在几个附近小群体内。然而, 通过邻近交叉, 区域的最优方法可以传播到整个群体<sup>[8]</sup>。

### 2.3 质粒迁徙和细菌基因演算法

根据已经观察到得细菌行为, 有微生物基因演算法、细菌演算法、伪细菌基因演算法、侧面基因转移以及质粒迁徙等等几种基因演算法存在。这些演算法回避了有性繁殖, 所以它们不能归为平行基因演算法, 尽管它们分布非常广。下面描述两种演算法, 它们将是本文演算法的基础。

质粒迁徙 (PM) 是以将某些细菌的行为为基础的。质粒是自我复制的染色体外 DNA 分子, 它们对于细菌的生存并不是必须的, 但它们含有各种基因株使细菌在多样的环境中更好的生存。因为健康个体可以在媒介中存储质粒, 不够健康的个体就可以从媒介中获取质粒, 因此质粒都可以在同代细菌中转移。这种能力使得细菌对于环境变异有很强的适应性。这可以类推到基因演算法中, 通过基因转移运作, 把已适应的个体中的优秀特质传播到整个群体。

细菌演算法中, 染色体被分成  $P$  部分, 每个部分产生自身的  $m-1$  克隆体。随机挑选的其中第  $i$  部分发生变异, 其最适合的部分被复制到  $m$  个体中。此后, 变异—评估—选择—替代过程在所有  $P$  部分中重复, 最适个体进入了下一代而其他  $m-1$  个个体死亡。

质粒迁徙和细菌演算法都是贪婪演算法, 在每一步都作出可见的最佳决定而不考虑将来的后果, 所以, 只能产生区域性最佳解决方法。相比之下, 这些方法可以很快找到, 牺牲最佳性的同时增强了适应性。然而, 在很多情况下, 在一个精心设计的质粒迁徙算法中, 个体的流动性使得优秀质粒在整个群体中传播, 因此, 通过在解空间里更加完全的搜索可以找到更好的解决方法。本文将提出一个强化的 cPGA 演算法, 这一方法包括一些贪婪细菌所给的启发, 来获得快速汇聚性、最佳性和适应性。

### 3 合作的在线分布演变模式

为了提高汇聚速度和适应性, 本章设计了信任评估系和策略演算法, 而且信任评估和策略演变都

是在分散的网络节点上进行。在该信任模式中, 节点间的交流以随机配对游戏中的重复囚徒困境为基础。每个中介节点采用某种策略判断某源节点发送来的数据包应该转发还是丢弃。这种策略取决于两点: 这一中介节点作为源节点时的网络既往行为, 以及这一中介节点对源节点的信任水平。这一模式包括: ①信任评估机制; ②以游戏为基础的网络模式; ③策略; ④以质粒迁徙为基础的区域性基因演算法, 用于发展高度分散情境下的策略。

合作的在线分布演变模式参考了集中模式。从本质上说集中模式和合作的在线分布演变模式的相同点在于策略模式和策略编码, 这些相同点使性能更加容易。然而两种信任评估机制和基因算法是完全不同的。

#### 3.1 信任评估机制

每个节点依据已观察到的邻居的行为都保持有一个信任表。例如, 如果节点  $B$  在观察传输范围内的节点  $A$ , 它就可以知道发送给  $A$  转发的数据包数量  $n$ , 以及  $A$  实际转发的数据包数量。于是  $B$  就可算出  $A$  的转发率, 方法如下:

$$f_r(B, A; n) = \frac{n_A}{n} \quad (1)$$

用简单的累计平均计算出转发率。比如,  $B$  可以通过观察  $A$  是否转发了第  $n$  个数据包更新  $A$  的转发率。用这一公式:

$$f_r(B, A; n) = \frac{1}{n} \sum_{i=1}^n d_i = \frac{(n-1) f_r(B, A; n-1) + d_n}{n} \quad (2)$$

其中  $d_i \in (0, 1)$  由第  $i$  个观察点决定。

既然策略可以被节点不断更新来适应环境变化, 那么长期的观察记录就有失公平, 因为早期的决定可能是在不同于现行决策的另一决策下做出的。为此, 本文决定只考虑观察到的最近的  $m$  个决定, 所以本文计计算的转发率是流动的平均值, 即在收到的前  $m$  个数据包中被转发的比例。

存储祁度值  $m$  要有一个权衡: 既要  $m$  足够大获取公平的转发率, 又要  $m$  足够小确保转发率是对应当前策略。用受限的存储祁度值  $m$  计算转发率, 要求先前的转发率和后  $m$  个决定是状态变量, 如下 ( $n \leq m$ ):

$$f_r(B, A; n) =$$

$$\begin{cases} \frac{1}{n} \sum_{i=1}^n d_i = \frac{(n-1) f_r(B, A; n-1) + d_n}{n} \\ \frac{1}{m} \sum_{i=0}^{m-1} d_{n-1} = \frac{m f_r(B, A; n-1) + d_n - d_{n-m}}{m} \end{cases} \quad (3)$$

根据当时的转发率  $f_r(B, A; n)$ ,  $B$  可以决定对  $A$  的信任水平  $T\{B : A; n\}$ , 如表 1 所示。为了比较, 在集中模式里, 保持同样的转发率和同样的信任值。

表 1 信任评估机制中转发率和对应的适应度  
Table 1 Fitness of forwarding rates and correspondence in trust evaluation mechanism

$f_r(B, A; n)$	0.9–1.0	0.6–0.9	0.3–0.6	0.0–0.3
$T\{B : A; n\}$	3	2	1	0

最后, 必须提到的是, 只用参考前  $m$  个决定的流动平均值也许只是一个细微的细节, 但本文引入它是为了决策的在线适应性。

### 3.2 基于游戏的网络模式

基于游戏的网络模式即为以一次游戏作为一次网络行为, 每个收取数据包的中介节点应该采用一策略决定转发还是丢弃。每个游戏开始于源节点发送数据包, 结束于数据包被送至终点或被中介节点丢弃。一旦游戏结束, 依据其决定和源节点的信任水平, 每个参与者收到一个回报。因为有两种节点, 源节点和中介节点, 所以要保持两种回报表, 一个节点的信任水平, 往往决定它在这个网络中的价值, 中介节点需要根据源节点的价值来决定是否转发。至于一个源节点他的价值(信任水平)如何体现, 由两个方面决定, 一是作为原节点时, 如果它的信息能够被转发成功, 那么说明它的信息很有价值, 对网络有积极作用, 奖励 5 分, 不成功, 浪费了网络资源, 不得分。二是作为中间节点时

候, 如果源节点的信任水平高, 它转发得分就高, 丢弃得分就低, 如果源节点信任水平很低, 还转发, 那就得分很低, 不转发得分反而高, 这是一种相互牵制, 相互激励的方法, 如表 2 所示。

表 2 基于游戏的网络模式中源节点和中介节点对应的回报值  
Table 2 Return value of source node and intermediate node in Game-based network model

$T$	3	2	1	0
转发	3	2	1	0.5
丢弃	0.5	1	2	3

为了方便比较, 这些回报直接从几种模式中提取, 同时也因为它们有两个优点: 一次成功的传输是最有价值的事件, 而且关于信任值, 丢弃或转发之间是相称的, 表明对于每个节点, 节约节点的能量和获取邻居信任同等重要。

一个中介节点的转发或丢弃决定始终被沿路的所有节点观察。而且, 收到数据包的节点可以考虑它之前的所有节点都合作了。

### 3.3 策略

本文用一个类似于集中模式中所用方法的策略整编码法。节点作为中介节点时所遵循的策略由位串编码, 其中每个比特代表丢弃 (D) (bit=0) 或合作 (C) (bit=1) 的决定。这一策略取决于如下参数(此节点对源节点的信任水平): 前两次游戏中此节点作为来源时的传播状态: 成功 (S) 或失败 (F)。

结果策略有 16 位比特, 如表 3 的策略举例所示。例如, 根据表 3 的策略, 如果它对源节点的信任水平为 1 而且前两次自己作为源节点的数据传送是成功的, 中介节点就会发送数据包(从左向右数第 8 比特)。

表 3 编码策略例  
Table 3 Example strategy code

编码策略, 策略例: 0001 0011 0101 0111															
源节点信任水平	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3
转发状态 2	F	F	S	S	F	F	S	S	F	F	S	S	F	F	S
转发状态 1	F	S	F	S	F	S	F	S	F	S	F	S	F	S	S
决策	D	D	D	C	D	D	C	C	D	C	D	C	D	C	C

策略通过基因演算法发展。可能是先随机挑选一个初始策略然后开始演变。但是, 如果初始策略的某些比特设置为特定的值, 达到最优化合作的集合速度就可以提高。另外, 假如节点作为源节点还不足两次, 它们也会合作。

### 3.4 分散的细菌式进化演算法

基因演算法可以最大化适应性或每个节点的平均回报值。为设计演算法, 重复几个步骤, 探测几种设计可能, 来平衡研究过程中出现的不同问题。下面将介绍演算法及其设计。并且, 在后面部分,



会清楚展示设想在合作程度、汇聚速度、节能性和适应性方面优于集中演算法。

在分散细菌式进化演算法中, 随网络进程在线即时进化策略。在这一网络进程中, 一连串游戏发生。一个游戏就是一次或成功或失败的数据包传送。在一次游戏中, 源节点在  $r$  条可能的路线中选择最信任的  $h$ -hop 路线发送数据包。数据包传送到目的地即游戏成功。当某节点作了  $R$  次源节点, 就称它完成了一个质粒迁徙周期 (PMP)。这时, 进化步骤必然产生, 也就是, 该节点和其邻居交换基因信息并进化策略。

让每一节点从所有 "一跳" 邻居中接收基因信息以开始繁殖并建立新的策略, 把质粒迁徙概念和传统的细胞基因演算结合在了一起。关于繁殖, 本文用了传统的单点交叉和变异过程。通过允许每个节点保留自身最好的先前策略, 引进了细胞质粒迁徙, 因此, 如果在现有的质粒迁徙周期中, 新策略并不能增加适应性, 可以恢复原策略。该模式的运行方式在下面的演算法 1 (Algorithm 1) 中有详细展示。

#### 演算法 1

```

1 begin
2 si (0) ←GenerateStrategy (); //Generate the
3 initial strategy
4 j←0; //Time index, expressed in PMPs
5 While ture do
6 fi (j) ←GetFitness (si (j) ); //Evaluate this
7 strategy (play some games)
8 if (j>0) &&. (fi (j) <fi (j-1) ) then //
The previous
9 strategy was better
10 si (j) ←si (j-1); // Restore the previous
strategy
11 fi (j) ←fi (j-1); //and the previous fitness
12 end
13 k←GetNeighborhood (i)
14 p1←Roulette wheel (k ∪ {i} ); // Get first
parent
15 p2←Roulette wheel (k ∪ {i} ); // Get first
parent
16 while p1==p2 do
17 p2←Roulette wheel (k ∪ {i} ); // we need
two
18 different parents

```

```

19 end
20 if fi (j) ≥Mean (fp1 (j); fp2 (j) ) then
21 si (j+1) ←si (j); //changing the strategy
22 is not worthy
23 else
    Si (j + i) ← Crossover (Sp1 (j) + Sp2
(j) );
    Si (j+i) ←Mutation (si (j+1) );
End
J++
End

```

#### 演算法 2

```

Pesudeo - code for randomly generating a new
strategy
GenerateStrategy {
S←null; //strategy to be returned
For (bit=4; bit<14; bit++) do
R←Random (); //0 or 1, each with
Probability 1/2
SetStrategyBit (s, bit, r); //set the
Given bit of s randomly
end
s←OR (s, 0000000000000111); //conditions for
initial collaboration
s←AND (s, 0000000000000111); // conditions
for
initial discarding
return s;
}

```

关于随机的初始策略, 与演算法 1 (Algorithm 1) 第 2 行的产生策略 GenerateStrategy 功能相对应, 本文对有些比特做了先天决定来加速演算法的汇聚, 如演算法 2 (Algorithm 2) 所示。本文在初始策略中特别固定了 6 位比特: 中介节点在源节点有最高信任水平且网络已至少成功发送前两次数据包之一时会予以合作 (即, 根据表 3, 14、15、16 比特被设置为合作——演算法 2 (Algorithm 2) 第 7 行)。同样, 如果发来数据包的原文件信任水平最低, 而且网络上前两次数据发送至少有一次失败, 那么中介节点会丢弃该数据包 (即, 1、2、3 比特被设置为丢弃——演算法 2 (Algorithm 2) 第 8 行)。本文在不同的模拟情形的最终的策略中都发现了这一模式, 所以本文这些假设已经得到验证。最后留下的 10 位比特是其 1/2 (演算法

(Algorithm 2) 第 3—6 行)。

节点在网络运行开始阶段即它作为源节点发送数据包不足两个时如何表现是一个需要处理重要的问题，因为当时没有可得到的网络行为信息。在这种情况下，认定了节点应该尽力立即开始在邻居中建立好名声，正如 Axelrod and Dion 所提。因此，当节点还发送不足两次数据时，中介节点的决策始终是合作，不论其源节点的信任水平如何。

另外本文对实验数据进行了一些简单的模拟，在模拟平台中，网络由一定数量的流动节点构成，它们有两类：标准节点和自私节点。这一环境（其特点是自私节点的比例），可以被设计成在模拟过程中随着自私节点和标准节点的相互转化而不断变化。网络模拟随着游戏的进行或数据包在任意源节点和目标节点间的传输而进展。对于每一组源节点和目标节点，会建立几条随意选择的路径。在一次游戏中，路径选择后，源节点发送数据包，各中介节点根据自己的策略选择转发或抛弃该数据包。某中介节点的转发或抛弃数据包的决定被途中的所有节点观察。如果数据包到达目的地，传输（或游戏）成功；如果数据包被某中介节点抛弃，传输（或游戏）失败。经过指定次数的游戏后，邻近节点间发生基因信息交换产生一次进化。这样可以观察到随程控环境改变而进化的策略。

这种流动性是通过两个历程完成的：在每次质粒迁徙周期中给每一节点随机选择一组邻近节点，和给每一游戏随机选择路径。在第一个历程中，在质粒迁徙周期，给每一节点随机选择一组 6 个邻近节点，在这些邻近节点间进行进化。这一邻域大小的选择是为了模仿典型的 6 边几何体：在每个顶点和中心都有节点。在第二个历程中，当数据包在源节点和目标节点间传输时，路径长度根据条件概率质量函数  $P_h$  来随机选择。假设路径长度为  $h$ ，路径数量  $r$  是根据条件概率质量函数  $P_{r/h}$  来随机选择的。为了性能评估和比较，这两种分布见表 4。在已知的路径中，源节点会选择最信任的路径，即依据自己的观察，选择转发率最高的中介节点参与的路径。

表 4 路径跳数  $P_h$  的概率分布和既定长度  $P_{r/h}$  的已知路径数  
Table 4 Probability distribution of path hops ( $P_h$ ) and number of discovered paths given the length ( $P_{r/h}$ )

	2	3	4	5	6	7	8
$P_h$	0.4	0.3	0.1	0.05	0.05	0.05	0.05
$P_{1/h}$	0.5	0.5	0.6	0.6	0.6	0.8	0.8
$P_{2/h}$	0.3	0.3	0.25	0.25	0.25	0.15	0.15

$P_{3/h}$	0.2	0.2	0.15	0.15	0.15	0.05	0.05
-----------	-----	-----	------	------	------	------	------

对于进化算式，交叉点建在第 8 比特，它标志信任节点和非信任节点的策略区别。如果发生交叉，每一比特的策略突变概率为 0.001。此处所示的所有模拟结果均是既定实验被平均独立复制 60 次后而得出的。

## 4 结 论

用染色体的交叉和变异，同时允许质粒保留诸如对前策略的恢复或子代拒绝亲代基因物质的行为。这种进化演算法的设计能在固有误差（其策略离最优不够近）和变异误差（其策略变化多端以致忘了优秀解决方法）中找到一个很好的权衡。这一过程很有挑战性，因为在本文的问题中有双重标准：需要同时获得正常节点间的高合作水平（最大化生产能力）和自私节点的有效隔离（最小化能量消耗）。很多设计都可以产出最优合作值，但其中的自私节点没有被彻底隔离；本文研究过的其它非常节能的设计又牺牲了正常节点的合作性。演算法 1 是本文的设计程序的产物，用附加的快速汇聚性和良好适应性等要求来衡量这些标准。而且，对解空间的延长探究是因为流动性和相邻重叠。这点确保了最优合作值的汇聚。

因此，本文的进化演算法最首要的特征是它的分散实施和合作汇聚速度。这些特点使演算法在容迟网络中可实施，因为采用了两种方法开发网络的自然结构：一是在本地用分散的方式实施进化；二是流动性使相邻处高度重叠，基因信息在几个质粒迁徙周期（PMP）后快速在整个网络中散播。

## 参考文献：

[1] 郭航, 王兴伟, 黄敏, 等. 容延容断网络研究及进展 [J]. 计算机科学, 2010, 37 (11): 12—18.  
[2] 邵清. 基于生物网络的延迟容忍网络容错研究 [D]. 上海: 东华大学, 2010.  
[3] 侯君婷. 简析 DTN 网络与传统网络的区别 [J]. 电信快报: 网络与通信, 2010 (4): 44—46.  
[4] 任延平. 基于基因算法的 MIMO—OFDM 移动通信系统的信道估计 [D]. 大连: 大连海事大学, 2010.  
[5] 高顺喜, 张龙. 容迟与容断网络的研究进展 [J]. 电信科学, 2010 (11): 56—65.  
[6] 郑炜. 延迟容忍网络的相关问题研究及仿真 [D]. 上海: 上海交通大学, 2007.  
[7] 张帆. 延迟容忍传感器网络性能研究 [D]. 武汉: 华中科技大学, 2007.

[8] 郑金华, 蔡自兴. 对等式网络并行 GA 的设计与实现 [J]. 计算机应用研究, 1999 (8): 33—34.