

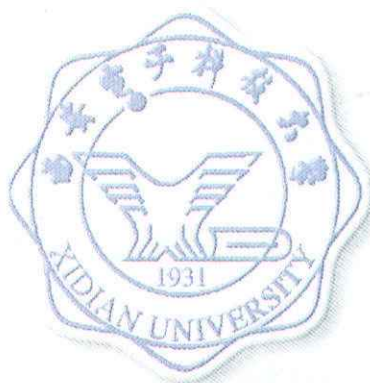
代 号 10701

学 号 0122420802

分类号 TP393.01

西安电子科技大学

硕士学位论文



题 (中、英文) 目 智能多播路由算法

Intelligent Multicast Routing

作 者 姓 名 杨海潮 指导教师姓名、职务 刘芳 教授

学 科 门 类 工学 学科、专业 计算机应用技术

提交论文日期 二〇〇四年一月

创新性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：杨海潮

日期 2004.1.4

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。本人保证毕业后离校后，发表论文或使用论文工作成果时署单位名称仍然为西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密的论文在解密后遵守此规定）

本人签名：杨海潮

日期 2004.1.4

导师签名：刘芳

日期 2004.1.4

摘 要

在计算机网络中，多播是目前研究最多、应用最广的连接方式。实现多播的一般方式是建立多播树，多播树是根为源节点，且覆盖所有多播成员的一棵树。多播路由算法主要就是用来建立一棵性能较好的多播树。

本文在新的智能计算方法—免疫克隆算法的基础上，给出了三种解决时延受限的多播路由问题的算法。主要内容如下：

1、提出了基于克隆策略的多播路由算法，将多播树的构造问题转化为备选路径的选择问题，再利用克隆策略来处理备选路径的选择。仿真表明，该算法有效的克服了早熟现象，而且算法稳定、灵活，操作简单。

2、结合免疫算法和克隆策略的优势，提出了免疫克隆多播路由算法。与传统的遗传算法相比，本算法通过引入与具体问题相关的先验知识来指导搜索过程，加速算法的收敛速度。

3、提出了参数可调的克隆多播路由算法，该算法在克隆多播路由算法的基础上引入一个动态可调因子来控制算法的搜索空间，加速了算法的收敛速度。

关键词：多播路由 智能计算 免疫算法 克隆策略 服务质量

ABSTRACT

In computer networks, multicasting is becoming increasingly important. Tree construction is a commonly used approach arises from the ability to potentially share many links in transmitting the message to destination set. Multicast routing algorithms are used to compute a multicast trees of good performance.

Based on these new Intelligent algorithms , three delay-constrained multicast algorithms are proposed. The main research work and results are followed:

1、Clonal Strategies based multicast algorithm is proposed. It converts delay-constrained multicast problem to the selection of candidate paths, and uses Clonal Strategies to select candidate. Simulations prove, it saliently solves the “prematurity” problem and therefore has better ability of global searching with the property of stabilization, agility and operating simply.

2、Immune Clone Based Multicast Routing is proposed by combining the advantages of Immune Algorithm and Clonal Strategies. Comparing with the conventional genetic algorithm(GA).The algorithm instructs the search process and accelerates the search speed by importing the transcendental knowledge relative to some specific problems.

A Clonal Strategies(CS) based multicast algorithm is proposed in this thesis. the algorithm is accelerated by using an adjustable parameter based on clone multicast algorithm to control the search space.

Key Word : Multicast, Intelligent Computing, Immune Algorithm , Clonal Strategies, QoS

目 录

摘要

ABSTRACT

第一章 绪论.....	1
1.1 多播技术的产生背景.....	1
1.2 多播技术概述.....	1
1.3 多播的特点.....	2
1.4 多播算法.....	3
1.4.1 动态算法和静态算法.....	3
1.4.2 分布式算法和集中式算法.....	3
1.4.3 分层多播路由算法.....	4
1.5 多播路由协议.....	4
1.5.1 多播开放式最短路由优先协议 (MOSPF Multicast Open Shortest Path First) ...	5
1.5.2 距离向量多播路由协议 (DVMRP Distance Vector Multicast Routing Protocol)	5
1.5.3 独立点到点协议 (PIM Protocol Independent Multicast)	5
1.6 本文的内容概要.....	6
第二章 多播树理论基础及基本算法.....	7
2.1 Steiner 树问题的定义.....	7
2.2 多播树启发算法.....	8
2.2.1 无约束最短路径算法.....	8
2.2.2 无约束最小生成树算法.....	9
2.3 QoS 的基本概念.....	10
2.3.1 服务质量的基本概念.....	10
2.3.2 基于 QoS 的路由选择机制的基本假设.....	11
2.3.3 QoS 多播路由问题的数学模型.....	12
2.4 智能多播路由算法.....	13
2.4.1 遗传算法.....	13
2.4.2 基于遗传算法的多播路由算法.....	15
2.5 随机网络产生模型.....	17
第三章 克隆多播路由算法.....	20
3.1 引言.....	20
3.2 免疫克隆选择算法.....	20
3.3 时延受限的多播路由问题的描述.....	22
3.4 启发式算法.....	23
3.5 克隆多播路由算法.....	24
3.5.1 符号约定.....	24
3.5.2 算法描述.....	26
3.5.3 算法复杂度.....	27
3.6 仿真实例及结果分析.....	28
3.7 结论.....	30
第四章 基于免疫算子的克隆多播路由算法.....	31
4.1 引言.....	31
4.2 免疫算法.....	31

4.2.1 产生背景	31
4.2.2 基本概念	32
4.2.3 免疫概念	33
4.2.4 算法描述	34
4.2.5 免疫算子的机理与构造	35
4.3 基于免疫算子的克隆多播路由算法	36
4.3.1 符号说明	36
4.3.2 算法描述	36
4.3.3 算法复杂度分析	38
4.4 仿真实验	38
4.5 结论	40
第五章 参数可调的克隆多播路由算法.....	43
5.1 引言	43
5.2 参数可调的克隆多播路由算法	43
5.2.1 算法描述	43
5.2.2 算法复杂度分析	46
5.3 仿真实验	47
5.4 结论	49
结束语.....	50
致谢.....	52
参考文献.....	53
作者在读期间撰写的论文.....	60

第一章 绪论

1.1 多播技术的产生背景

从上个世纪末到本世纪初是网络技术大发展的时代，特别是以因特网为代表的 IP 网络更是日新月异。因特网上的用户数量持续呈爆炸性增长，网上的应用由传统的电子邮件转向 FTP、WWW 等应用。此外，基于因特网的新应用和业务也在不断地推出，如电子商务、IP 电话、视频会议等。

但是，要满足这些业务的需求，特别是要保证一些实时业务的带宽、时延等特殊需求，以目前因特网中的尽力而为的服务是难于完成的。尽力而为服务的特点是对所有应用提供同种数据传输服务，对网络的资源缺乏有效的分配和管理，当网络负载较轻时，各个应用得到的传输服务质量尚可，但是随着用户数目的增多，网络的负载也将增加，此时，各种应用的行为表现为无序地竞争网络资源，造成网络资源的不合理占用各种应用各为其利，其结果是服务质量互相恶化。目前的因特网技术急需进行改进以提供有效的资源分配与管理。

前面提到的新型网络应用，如视频会议、交互式游戏、声音/视频电话、实时多媒体播放、分布式计算、视频点播和远程教学等应用的特点是涉及多个成员的交互，本质上具有多播的特征，与单播不同，如果网络不提供合适的协议支持，这些应用实现起来代价很大或不可行。所以，这些应用在功能和性能两个方面对传统网络协议提出了新的挑战。

1.2 多播技术概述

所谓多播，指的是一个源节点向多个目的节点发送信息的通信方式，参与多播的多个目的端点组成了一个多播组，每个端节点称为多播组成员。实现多播功能可以有 3 种方式：一是由源节点向各用户单独发送同样的消息，这种方式以单播为基础，因此每次会话需要产生与用户数量相等的分组，用户数量较少时，基本可以满足需求。二是每个分组含有一张目的节点清单，当分组到达路由器时，路由器检查所有目的节点，以确定将要用到的输出链路的集合。路由器为所使用的每一条输出线路复制一个新的分组，每个分组中仅含有使用此线路的那些目的节点。这种路由方式就像各自分组寻址一样，只是若干分组必须经过同一路由时，其中一个分组负担全部费用，而其它的分组不必承担费用^[1]。三是建立多播树，由源节点向树中的相邻节点发送消息，然后这些相邻节点继续向树中下一级节点转发这些消息。这样大大减少了源节点发送的消息量，防止了网络和链路的阻塞。

多播树分为两种类型^[8]：有源树和共享树。有源树的根是信息源，叶子节点必然是信宿节点。有源树的计算分为两种：第一种是最短路径树，从源节点到各多播组成员都以最短路径相连，用的较广的两种算法是Bellman-Ford和Dijkstra^[9]，它们分别是距离矢量和链路状态协议的基础。很明显，最短路径树的全树费用并不一定就是最优的。第二种有源树是最小Steiner树，即：在一个连通图中，给定一个节点子集，要求在图中找到一棵覆盖给定子集的费用最小的生成树。这是一个NPC问题，即最优算法无法在多项式时间内完成。因此，要寻求有效的启发式算法，降低算法复杂度，而性能上逼近理论算法。目前Steiner树问题的算法很多，Hwang在文献^[3]分别提出了精确和启发式的Steiner算法，其它的一些较为有效的算法还可以从文献^[4~7]中看到。有源树的缺点在于，对于有m个成员的多播组，则需要生成m棵多播树，其计算和存储费用都很大。

共享树，也称为核心树，最早在文献^[10]中被提出，只需要为每一个多播组建立一棵多播树，各组节点之间的通信通过被称为core^[11]（或RP:Rendezvous Point）的节点进行：各节点通过最短路与核心相连，将消息发往核心，再由核心发往接收节点。可以看出，与有源树相比，共享树减少了很大的计算和存储开销。但一般共享树的费用比有源树要大2倍以上，如果核心是组成员的话，其代价会达到有源树的3倍以上^[12]，现有的路由协议有protocol independent multicast (PIM)^[13]和core-based trees(CBT)^[14]。共享树的问题主要在于核心的选取和维护较为困难，文献^{[12][15]}中给出了几种选取核心的方法。此外，为降低中心失败带来的影响，通常要维护一个包含多个核心的列表。

1.3 多播的特点

多播的特点主要体现在以下三个方面：

1. 带宽

运用多播技术分发信息常常能从本质上减少整个网络带宽的需求。与单播技术（点到点通信方式）相比，当多个用户要求同一服务器提供同样信息时，使用单播技术，带宽将随用户的增多而增加需求，而对于多播，由于在共用链路上传递信息的一份拷贝，因此带宽的需求不会随用户数量的增加而增加。

2. 服务器负载

对此我们举例说明，一家公司正在使用一种音频服务器传送一个无线节目，在互连网上实时连接用户。如果此公司使用单播传送机制，它将需要不断增加它的实时音频服务器的能力和数量以满足连接用户的增长。服务器必须为每一个收听节目的用户发出各自的信息流。当连接的用户数目增加时，服务器负载也增。

当服务器负载增加到一定程度，服务器就不能再发出信息流。另一方面，如果使用多播来发布节目，只有单个实时数据流要成为连接所有用户的节目源头。用这种方式，公司不需要购买越来越多高性能实时音频服务器以满足用户数目的增长。很明显多播提供的主要优势在于降低了所需服务器性能。

3. 网络负载

当将相同的内容传送给多个用户时，多播能明显地减少带宽要求，带宽消耗的降低等同于路由器上的负载降低。通常本假定是真实的，但在一些情况下，在某些特定点路由器的工作负载可能会加大。请再参考一下上节的例子，我们知道第一跳路由器（直接与服务器相连的路由器）从服务器接受一个数据流。然而要注意的是，第一跳路由器把单个数据流复制成两个输出数据流以便将该数据送交给下游用户。这个复制过程成为附加在路由器上的工作负载，它需要在整个网络设计中加以考虑。如果一种路由器没有一个有效的复制机制，则当输出接口数很大时该路由器负载将明显增加。

1.4 多播算法

为进行有效地多播通信，确定多播路由是非常关键的。多播路由算法就是用来确定多播树。并且一般用树的“费用”来衡量多播树的好坏。多播树的费用是指树中所有链路费用的总和^[2]。在实际应用中，一般要有效利用网络资源，为此要求多播树的费用最小。

1.4.1 动态算法和静态算法

多播路由算法按其研究网络的组成员是否变化可以分为静态和动态算法两类。静态多播路由算法针对初始多播组成员构造一棵多播树，它不能根据当前实际（实测或估计）传输量和拓扑变化来做拓扑选择，而是按初始设计好的路由传送，它的路由修改经过一定运行后才能进行。但是真实网络中存在许多动态变化。如网络拓扑结构的变化，组成员的变化等。而动态多播算法就可以处理成员的加入和离开之后多播树的更新问题。

1.4.2 分布式算法和集中式算法

多播路由算法按其实现方式，又可分为分布式和集中式算法，集中式路由算法是由节点在掌握整个网络的拓扑结构后，确定多播路由。集中式多播路由一般都是源路由算法（source-rooted routing），即源节点通过某个链路协议获得完整

的网络拓扑信息，进行路由的计算。而分布式算法不需要每个组成员掌握整个网络的拓扑。每个组成员在只具有局部信息条件下就可参与路由的确定，这对实现大型网络的多播路由是十分有利的。

1.4.3 分层多播路由算法

随着网络规模的扩大，路由器的路由表也会成比例增长。增大的表格不仅占用路由器的内存，而且需要更多的 CPU 时间扫描表格，以及更大的带宽来发送关于表格的状态报告。在某一时刻，网络可能会增大到不可能让每个路由器都给出至其他每个路由器的路径表项。为此出现了分层路由选择机制，分层路由算法是在将网络先按不同等级，不同层次划分，每个路由器知道在自己的区域内怎样选择路由和如何将分组送到目的端的全部细节，但并不知道其他区域的内部结构。采用分层路由算法有许多集中式算法无法相比的优点，如：

(1) 由于采用分层的方法，顶层的节点数大大减少，由此减少了路由表的搜索时间。路由表的计算量也大大减少。每一层节点内部的路由表只在内部有效。

(2) 每一层的节点只保存当前层的状态信息，与上下层都没有关系，从而减少了对内存的需求量。

(3) 每一层节点只在同一层内进行节点状态信息的交换，从而降低了信息的传输量，节省了网络资源。同时，本层节点内部状态信息的交换被限制在节点内，不会扩散到其他节点。

(4) 在每一个节点所代表的一个区域内，所采用的路由协议和路由算法与别的区域无关，也就是说，不同的区域可以采用不同的路由协议和算法。

分层路由机制的缺点是：

(1) 分层的处理，区域的划分，导致了网络状态信息的不准确、不及时。

(2) 网络状态信息的不准确，导致无法提供最优化路径，降低了寻径性能

采用分层路由机制，并不是说现有的路由协议和算法没有用处了。因为在一个区域内，节点数是很少的，局部网络的规模很小，完全可以采用小规模网络情况下的路由算法。这样，一方面可以研究分层寻径机制，另一方面可以继续研究小规模网络情况下的路由算法。

1.5 多播路由协议

要确定多播路由，首先要收集网络中的相关状态信息，路由算法就在这些信息的基础上来确定路由。网络的状态信息包括网络的拓扑结构、链路和路由器的

负载程度、链路和路由器的生效和有效、网络中路由器的类别（是否具有多播功能）等。这些信息的收集是由路由协议（Routing Protocol 来完成的。这一部分介绍几个实际应用中的多播路由协议，其中 DVRMP 和 MOSPF 是由 Internet 上的点到点路由协议 RIP（路由选择信息协议）与 OSPF（开放式最短路由优先）派生出来的。

1.5.1 多播开放式最短路由优先协议（MOSPF Multicast Open Shortest Path First）

MOSPF^[16]是利用点到点的链路状态数据库，以 OSPF 为基础的多播路由协议，由于 OSPF 应用 Dijkstra 算法进行路由选择，因此每个节点都要保存全网的拓扑信息，所有节点对网络的看法一致，它不需发送任何分组，节点就可根据全网；链路状态表计算组中每个信源的多播树，而且各节点对此树的看法一致。为了减少计算量，MOSPF 可以按需执行算法，即只有当一个节点收到一个信源关于某个多播组的第一个分组时，才执行算法。这种做法的缺点是对第一个分组带来较大延时。MOSPF 的最大优点是享有 OSPF 对网络拓扑的变动快速反应能力。然而这个能力是以对路由器 CPU 资源的巨大消耗为基础的，而且随着网络中组数量的增加，这种消耗也在迅速增加。

1.5.2 距离向量多播路由协议（DVMRP Distance Vector Multicast Routing Protocol）

DVMRP^[17]是 Mbone 上广泛运用的多播路由协议，它采用距离向量算法。在 Mbone 上采用隧道技术，使多播通信得以在支持多播的子网之间实现。由于 Mbone 的飞速发展，由 DVMRP 而导致的大量路由控制分组定期在网络中的扩散开销限制了网络规模的发展。为此提出了分层 DVMRP^[18]，按照区域分割的方式对 Mbone 进行多层管理，区域内多播可以按照任何协议进行，而区域之间的多播由边界路由器在 DVMRP 协议下进行。由于采用多播协议的层次叠加，减少了路由控制信息开销。

1.5.3 独立点到点协议（PIM Protocol Independent Multicast）

PIM^[13]设计的出发点在于在广域网范围内同时支持共享树与有源树，并能完成两者之间的灵活转换，因而集中了两者的优点同时避免它们的缺点，在组员密集时以广播形式传送数据，然后从树上删除不存在接收节点的分支；而组员分布稀疏时，构造共享树传送，避免分组的广播开销。相应 PIM 有两种模式：稀疏模式（SM: Sparse Mode）和密集模式（DM: Dense Mode）。

PIM-DM^[19]是指多播组所覆盖的区域内，具有该组用户的子网数量在子网总数中占很高的比例。PIM-DM 基本上与 DVMRP 相同，属于数据驱动型协议，路由算法构

成多播树，但协议只是直接使用点到点路由算法给出的路由表转发数据，因而独立于点到点协议。

PIM-SM^[20]的含义是：首先拥有组成员的子网数量远远小于 Internet 中网络数量，其次是组所覆盖的网络资源不足。在这种模式中，分组接收节点通过向指定点（RP:Rendezvous Point）发送明确的加入消息加入组中，数据发送节点通过 RP 公布自身的存在。然后数据再以 RP 为根的共享树中流动。

总的来说，DVMRP、PIM 和 CBT 适用于 Internet 上数据报的应用环境，在一个通信会话过程中 MC 分组的路由可能发生变化，此时算法的简洁性比树的费用更重要。

1.6 本文的内容概要

本文主要研究了时延受限的多播路由问题，即如何建立满足时延要求且费用最小的多播树，提出了三种时延受限的多播路由算法。

本论文的内容安排如下：

第二章主要介绍了本文所涉及到的基本理论。主要包括了 Steiner 树理论和 QoS 问题的简要介绍，进化计算的基础理论的概述。总结了以往的无约束多播路由算法，以及当前两种主要的进化多播路由算法。本章最后介绍了本论文所采用的仿真网络模型的产生方法，它是由北卡罗来那州立大学的 Salama 和 Reeves 在 Waxman 网络的基础上提出的，与 Waxman 网络相比，该网络模型能更好的反应用户对节点的平均度的要求，并且绝对不会有孤立的节点产生。

第三章总结了以往的延时约束多播路由算法，给出了延时受限的克隆多播路由算法，并进行仿真实验，和进化多播路由算法进行了比较。

第四章给出了免疫克隆多播路由算法，进行了仿真实验，并和 BSMA 算法以及进化多播路由算法进行了比较。

第五章介绍了参数可调的克隆多播路由算法。

最后总结了全文。

第二章 多播树理论基础及基本算法

上一章讲到，对于多播路由问题，一般要建立费用最小的多播树，即求解最优 Steiner 树。为此本章介绍有关 Steiner 树问题的理论，其中包括求解 Steiner 树问题的启发式算法；算法的复杂度、性能。最后，我们介绍了本论文所采用的仿真网络模型，由于其能较好地模拟真实网络，因此被大多数研究多播算法的学者所采纳。

2.1 Steiner 树问题的定义

建立最优 Steiner 树被认为是实现多播通信的最好办法之一，Steiner 树理论及其算法也成为求解多播树的基础。对 Steiner 树问题的研究早在 60 年代就开始了，到现在为止，已经有很多这方面的文献。Steiner 树问题的应用不仅在通信方面，在电路图的自动布线、交通线路的经济规划、车辆的调度和编组、通信线路的铺设等方面都有应用。Steiner 树问题有很多种类型，如网格 (grid) 上的 Steiner 树问题、Euclidean 平面上的 Steiner 树问题、一般连通图中的 Steiner 树问题等。下面首先给出无向带权连通图上 Steiner 树问题的定义。

首先将计算机网络抽象为无向连通图 $G = (V, E)$ ，其中 V 为网络节点（路由器）的集合， E 为边（通信链路）的集合，在每条边 $e \in E$ 上定义一个权值函数 $c(e): E \rightarrow R^+$ ，表示每条边的费用，给定一个节点子集 $Q \subseteq V$ ，这里 Q 对应为多播组成员，Steiner 树问题就是要寻找一棵覆盖给定节点集合 Q 且费用最小的最优 Steiner 树。

从最优 Steiner 树的定义可知，最优 Steiner 树中的叶子节点都应是 Q 中的节点，一般称 Q 中的节点为基本节点：最优 Steiner 树中除基本节点外的其它节点被称为 Steiner 节点，图 G 中所有可能成为 Steiner 节点的节点集合记为 S ，易知 $Q \cap S = \emptyset, V = Q \cup S$ 。

最小生成树和最短路问题都是 Steiner 树问题的特例。当 $Q=V$ 时，Steiner 树问题就转化为最小生成树问题，当给定节点集合中只包含两个节点时，即 $|Q|=2$ 时，就变为最短路问题。最小生成树问题和最短路问题都存在多项式时间的最优解法，但目前 Steiner 树问题不存在多项式时间的最优解法，它是一个 NP 完全问题。一个合理的目标是寻找该问题的一个启发式算法，使我们能在一个低阶多项式时间内得到一个“接近”最优的解，为此对 Steiner 树问题一般讨论解决它的启发式算法。启发式算法并不保证给出最优解，那么如何评价启发式算法的好坏呢？主要是基于两个方面的考虑：首先是时间复杂性方面的要求，即要求有一个多项式时间界；其次是性能方面的要求，即希望所求的近似解尽可能“接近”最优解。可

以从不同角度来评价启发式算法的性能，大体上可分为三类：第一类是以算法在最坏情况下的行为为标准，研究算法得到的近似解与最优解的接近程度，越接近越好；第二类是以算法的平均行为为标准，研究得到最优解的概率；第三类是局部搜索算法，寻找局部最优解，这种方法有时很好，有时很差，只能通过实践加以评定。对于算法最坏情况下的性能比，可以从理论上给出一个上界，但最坏情况下的性能不能说明算法的实际性能如何，因为在实际应用中，算法体现出来的是平均性能，最坏情况很少发生，而算法的平均性能是较难从理论上给出的，为此要通过仿真实验来研究算法的平均性能。

2.2 多播树启发算法

2.2.1 无约束最短路径算法

正如其名称，最短路径算法以分别最小化连接源节点和组成员的各条路径的长度为目标。最短路径树（SPT）的性质根据衡量链路的标准不同，分为以下几种：

- 1, 最小跳数（minimum-hop）树，每条链路都设为单位长度。
- 2, 最短路径费用（LC:Least-Cost）树，长度设为该链路费用，此时最短路径算法即LC算法，其目标函数可用下式表示。

$$\min(\text{cost}(P_T(s, g))) \quad \forall g \in Q$$

这里， T 是一棵根为信源节点 s 并且覆盖组成员集合 Q 的树，其总费用不一定是最优的。

- 3, 最短路径费用（LD:Least-Delay）树，目标函数如下式。

$$\min(\text{delay}(P_T(s, g))) \quad \forall g \in Q$$

其参数意义同LC，只是链路代价变成了时延（delay）。

Bellman-Ford 算法^[21]、Dijkstra 算法^[9]是两个最著名的最短路径算法，这两种算法都可以在多项式时间内收敛。Bellman-Ford算法的最差时间复杂度为 $O(|V|^3)$ ，这里 $|V|$ 指的是网络节点的总数。文献^[22]给出了分布式Bellman-Ford 算法。它的计算仅需要保留在各个网络节点中有限的网络信息即可。但Awerbuch等人^[23]证明了在最坏情况下分布式Bellman-Ford 算法的计算复杂度有可能随网络节点数的增加而呈指数级增长，因此提出了该算法的两种不同的分布式版本，有效的解决了计算复杂度过大的问题。Dijkstra 最短路径算法没有分布式计算的版本，它的最坏时间复杂度为 $O(|V|^2)$ 。文献^[24]给出两种算法运行时间的比较。这两种算法都能很

好的运行在非对称网络上。

另一种最短路径算法是由Dalal 和Metcalf^[25]提出的反向路径算法(RPF:the reverse path forwarding)。它首先由源节点向整个网络发出广播信息,当其它节点受到该信息后,则建立反向最短路径,源节点的信息就沿该反向路径发网各网络节点。RPF只能运行在对称网络上。Deering^[26, 27]将RPF应用于广播及多播问题提出了TRPB(the truncated reverse path broadcasting)算法和RPM (the reverse path multicasting) 算法。

2.2.2 无约束最小生成树算法

其目标函数如下:

$$\min(\text{cost}(T(s, Q)))$$

$T(s, Q)$ 表示覆盖源节点和组成员的一棵生成树。该问题是NP完全问题^[28],无约束Steiner算法并不能优化端到端时延,因此也许并不适合实时网络的应用。最著名的无约束最小Steiner树算法有Kou, Markowski, 和Berman提出的DNH(KMB)算法^[5], Takahashi 和 Matsuyama的TM启发式算法^[7], 以及Rayward-Smith的RS启发式算法^[6]。

KMB算法在计算过程中使用了Prim算法。由于Prim算法是被用来优化对称网络的,因此当被应用于非对称网络时,KMB算法的性能也会受到影响。KMB算法的最坏时间复杂度为 $O(|Q||V|^2)$,在文献^{[29][30]}中,Wall提出了KMB算法的分布式计算的版本。由KMB算法在对称网络上得到的多播树的费用一般只比最优Steiner树^[31, 32]要高5%。

TM算法构建多播树时,一开始令树中只包含源节点,然后逐次加入组成员节点。每次将形成的多播树与树外的一个组成员节点通过树与该节点之间的LC路径相连,直到所有的组成员都已经包含在树中才结束算法。TM算法的最坏时间复杂度为 $O(|Q||V|^2)$

RS算法以包含不同组成员的森林开始,逐个连接两棵距离最近的树,直到原来的森林只剩下一棵树为止。Rayward-Smith和Clare通过仿真证明^[33],RS算法得到的多播树的费用要比KMB算法和TM算法更接近最优结果。然而,RS算法开始也只是被设计用来解决无向网络中的问题的,目前还没有将RS算法实现与有向网络中的方法。

Jiang^[34]对KMB算法和RS算法提出了改进,改进后的算法比着原始算法得到的多播树的代价要小了很多,随后Jiang又提出了改进后算法的分布式版本^[35]。

最近,Ramanathan^[4]又提出了一种解决不对称网络的最优Steiner树的启发式

算法。相较前几种算法，该算法更为灵活，它通过一个可选参数 K 来平衡算法的执行时间和树代价，作者显示了Dijkstra最短路径算法，KMB算法，TM算法分别是参数 K 等于1， $(|G|+1)$ 和 $|V|$ 的情况。

其它的启发式算法参见Chow^[36], Leung 和Yum^[37], Bauer和Varma^[38]。

2.3 QoS 的基本概念

随着互连网络技术和网络应用的普及，多媒体业务将称为网络传输业的主流。而多媒体业务如连续视频、音频等实时信息的要求，使得网络技术的研究和发展面临新的挑战。在当前的 Internet 上，同一会话（session）的数据包可以沿不同的路径到达目的地。网络资源（如交换机缓冲区和链路带宽等）由来自不同会话的数据包共享。然而，这一结构不能满足需要承载多种数据类型的综合服务网络的要求。因为数据包可能会历经无法预测的延迟，并且以错乱的顺序到达目的地。这使实时信息的服务质量难以得到保证。因此，下一代高速广域网对实时交通来说应是面向连接，有质量保证的。

2.3.1 服务质量的基本概念

服务质量（QoS: Quality of Service）的概念^[39]被用来刻画服务提供者与用户间用数量或质量来定义的性能约定。一次连接的服务质量由一系列约束条件给出，如带宽约束、延时约束、抖动约束等。QoS 路由的基本任务是为一次连接寻找一条有足够资源、能满足 QoS 要求的可行路。另外，要尽力寻找使资源得到最优利用的可行路。QoS 路由不同与最大能力（best-effort）路由，前者通常是面向连接的，有资源预留功能，并提供有质量保证的服务；而后者可能是面向连接的也可能是面向无连接的，根据当前可获得的资源不同，服务质量有不同的变化。QoS 路由要比 best-effort 路由复杂得多，寻找有两个独立约束的可行路问题为 NP 完全问题。

QoS 路由选择的任务就是要从源节点到目的节点找到一条具有足够资源的路径来满足端到端服务质量，按照最近的研究热点来看，有关 QoS 路由选择问题可以分为两类，即基于源节点的路由选择和分布式路由选择，基于源节点的路由选择是指每个节点都具有全局的网络状态信息，源节点利用这些信息进行计算并找出可行的路径，全局网络状态信息将通过链路状态协议周期性的刷新；分布式路由选择式指每个节点只了解它的局部信息，并通过在节点之间周期性的交换控制信息找到可行路径。基于源节点的路由选择存在以下几个问题：(1) 全局网络状态信息必须频繁地刷新，否则不能正确地反映网络的变化，例如，可利用带宽会随着业务量的接入而经常变化，因此，对于大规模网络来说，需要大量的通信开销；

(2) 由于大量的通信开销和控制信息的传播时延问题, 基于源节点的路由选择方案只能提供近似的全局信息, 全局信息的不准确性可能造成 QoS 路由选择的失败;

(3) 由于链路状态协议本身的局限性, 因此任何一个单节点不可能接入所有节点和链路的详细状态信息, 尽管等级路由选择方案可以解决这个问题, 但是状态信息的组合却加重了信息的不准确性; (4) 考虑到 QoS 路由选择是基于每个连接的, 所以当涉及多个限制时, 可能造成源节点的计算开销过大。

分布式路由选择可以把路径选择的计算开销分配给源节点到目的节点之间的所有中间节点, 因此, 该方法可以缩短路由选择的响应时间。路由选择性能还依赖于状态信息的准确性, 因此, 这和基于源的路由选择存在同样的问题。分布式多媒体应用通常有比较严格 QoS 需求, 路由选择算法需要考虑应用的 QoS 需求, 研究基于 QoS 的路由选择机制^[40]既有理论意义也有应用意义。

2.3.2 基于 QoS 的路由选择机制的基本假设

用“流”这一术语来抽象从源节点到目的节点的通信, 代表其间通信的数据流总称。资源分配对流进行。对于点对点通信, 每条流涉及一个源节点和一个目的节点, 对于组通信, 每条流涉及多个源和目的节点。

对于组通信, 在寻找路由时, 主要考虑成员之间的资源共享, 将成员的最大资源需求作为整个组的资源需求, 这既比较符合组应用的实际情况, 也有利于提高网络资源的使用效率, 还有利于降低使用网络的费用。

将网络抽象成由节点和边组成的一张连通图, 图上的每个节点都配备有限数量的 CPU 和缓冲资源。当有一条流通过一节点时, 该流将占用该节点一定数量的 CPU 和缓冲资源; 同时通过一节点的不同流占用的资源量累加; 当一节点的可用资源量 (CPU 和缓冲资源) 不足以满足新流的资源需求量时, 将拒绝新流通过。图上的每条边都配备有限数量的带宽资源; 当有一条流通过一条边时, 该流将占用该边一定数量的带宽资源; 同时通过一条边的不同流占用资源量累加; 当一边的可用带宽资源量不足以满足新流的带宽需求量时, 将拒绝新流通过。流通过节点时有排队延迟、发送延迟、出错率 (可以认为是比特出错率), 通过边时有传输延迟、出错率 (可以认为是比特出错率)。同一流对所途径节点与边的资源需求是一致的。

基于 QoS 的路由选择的目的就是要在图上找出既满足流对节点的 CPU 资源、节点的缓冲资源、边的带宽资源的需求又满足流对端到端延迟、端到端出错率的要求的费用最低的路由。在进行基于 QoS 的初始路由的建立时, 不仅需要考虑连通性而且需要同接纳控制、资源预约、QoS 计费等机制相互配合。

2.3.3 QoS 多播路由问题的数学模型

设 $N(V, E)$ 表示网络，其中 V 表示网络节点集， E 表示双向链路集， $s \in V$ 为多播源节点， $M \subseteq \{V - \{s\}\}$ 为多播终点集， R_+ 表示正实数集， R^+ 表示非负实数集，对于任一链路 $e \in E$ ，定义4种度量^[41]：

延时函数： $delay(e) : E \rightarrow R_+$ ；

费用函数： $cost(e) : E \rightarrow R_+$ ；

带宽函数： $bandwidth(e) : E \rightarrow R_+$ ；

延时抖动函数： $delay_jitter(e) : E \rightarrow R^+$ ；

对于任一网络节点 $n \in V$ ，也定义4种度量^[41]：

延时函数： $delay(n) : V \rightarrow R_+$ ；

费用函数： $cost(n) : V \rightarrow R_+$ ；

延时抖动函数： $delay_jitter(n) : V \rightarrow R^+$ ；

包丢失函数： $packet_loss(n) : V \rightarrow R^+$ ；

则对于给定的源节点带宽函数： $s \in V$ ；终点集 M ， s 和 M 组成多播树 $T(s, M)$ 存在下列关系：

$$delay(p_T(s, t)) = \sum_{e \in p_T(s, t)} delay(e) + \sum_{n \in p_T(s, t)} delay(n)$$

$$cost(T(s, M)) = \sum_{e \in T(s, M)} cost(e) + \sum_{n \in T(s, M)} cost(n)$$

$$bandwidth(p_T(s, t)) = \min\{bandwidth(e), e \in p_T(s, t)\}$$

$$delay_jitter(p_T(s, t)) = \sum_{e \in p_T(s, t)} delay_jitter(e) + \sum_{n \in p_T(s, t)} delay_jitter(n)$$

$$packet_loss(p_T(s, t)) = 1 - \prod_{n \in p_T(s, t)} (1 - packet_loss(n))$$

其中， $p_T(s, t)$ 为多播树 $T(s, M)$ 上源点 s 到终点 t 的路径。本模型只考虑网络节点的包丢失率（因缓冲溢出），而忽略链路的包丢失率，这与实际情况相符。

定义 QoS多播路由问题：

网络 $N(V, E)$ 。多播源节点 $s \in V$ ，多播终点集 $M \subseteq \{V - \{s\}\}$ ，延时函数 $delay(*) \in R_+$ ，延时抖动函数 $delay_jitter(*) \in R_+$ ，费用函数 $cost(*) \in R_+$ ，带宽函数 $bandwidth(*) \in R_+$ 和包丢失率 $packet_loss(*) \in R_+$ ，寻找一棵多播树 $T(s, M)$ 满足：

延时约束： $delay(p_T(s, t)) < D_i$

带宽约束： $bandwidth(p_T(s, t)) > B$

延时抖动函数: $\text{delay_jitter}(p_T(s, t)) \leq DJ_t$

包丢失率函数: $\text{packet_loss}(p_T(s, t)) \leq PL_t$

费用函数: 在所有满足条件的多播树中, $\text{cost}(T(s, M))$ 最小。

其中 $p_T(s, t)$ 为 $T(s, M)$ 上源点 s 到终点 t 的路由路径, B 是带宽约束, D_t , DJ_t , 和 PL_t 分别是 t 的延时, 延时抖动和包丢失率约束, 本模型中假设所有多播终点的带宽约束相同, 而延时、延时抖动和包丢失率约束可互不相同。

已经证明, 存在两个不相关可加度量的QoS多播路由问题是NP完全问题^[42], 显然, 上述QoS多播路由问题属于NP完全问题, 其延时、延时抖动、费用为可加度量且互不相关。

2.4 智能多播路由算法

智能计算(Intelligent Computing), 也称为软计算(Soft Computing), 是由模糊集理论的创始人, 伯克利大学教授L.Zadeh提出来的^[43]。虽然至今没有一个统一的定义, 但可以这样来概括它。智能计算就是借用自然界(生物界)规律的启迪, 根据其原理, 模仿设计求解问题的算法。目前这方面的内容很多, 如: 模糊逻辑(FL, Fuzzy Logic)、粗糙集(RS, Rough Set)、人工神经网络技术(ANN, Artificial Neural Networks)、遗传算法(GA, Genetic Algorithm)、进化规划(EP, Evolutionary Programming)、进化策略(ES, Evolutionary Strategies)、模拟退火技术和群集智能技术等。

由于本文所提出的三个算法都会和遗传多播路由算法作比较, 因此有必要先介绍一下遗传算法的基本概念和计算过程。

2.4.1 遗传算法

遗传算法(GA: Genetic Algorithm)是在70年代初期由美国的Holland教授发展起来的^[44, 45], 它是一类以达尔文自然进化与遗传变异理论为基础的求解复杂全局优化问题的仿生型算法。它模拟生物进化过程, 通过向自然学习来求解问题。

遗传算法利用简单的编码技术和繁殖机制来表现复杂的现象, 从而解决非常困难的问题。特别是由于它不受搜索空间的限制, 不要求诸如连续性、导数存在和单峰假设, 以及固有的并行性, 遗传算法已被广泛地应用于组合优化、机器学习、并行处理、人工智能、自适应控制、人工神经网络训练等优化问题中。它尤其适用于处理传统搜索方法难于解决的复杂和非线性问题。遗传算法是一种基于自然选择和遗传的概率搜索算法。遗传算法从一个称之为群体的随机初始解的集合开始, 通过对群体施加遗传操作实现群体内个体结构重组的迭代过程。下面,

介绍遗传算法中的一些最基本的概念。

1. 染色体 (chromosome) 又称为个体 (individual), 是生物学中的概念。在遗传算法中的染色体通常用一个所谓的串来表示, 如 $X = x_1x_2 \cdots x_p$, 其中 x_i 是串 X 的基本单元, P 称为基因 (gene) 数, 或者串的长度。在遗传算法中, 串一般根据问题的不同, 可分为二进制的0、1串, 整数串、实数串等。

2. 群体 (population) 个体的集合, 在使用遗传算法求解问题时, 首先随机地选择若干个个体构成一个群体。

3. 遗传算子 遗传算子作用在基因串上, 常用的遗传算子有如下三种:

(1) 选择算子 (selection operator)

选择算子是用来选择群体中适应度高的个体以构成新一代群体。其中每个个体被选中的概率 $p_i(X_i)$ 取决于个体 X_i 对环境的适应值 $f(X_i) = f_i$, 最常用的选择方法是适应度比例方法; 也称为赌轮选择。

(2) 交叉算子 (crossover operator)

交叉算子通常作用在两个基因串上。交叉算子是把两个基因串中的某一部分相互互换, 产生两个新的个体。交叉部分的选择是随机的 因此交叉算子是以一定的概率发生作用。交叉算子可分为单点交叉、多点交叉、均匀交叉、线性交叉和算术交叉等。

(3) 变异算子 (mutation operator)

变异算子是在遗传算法中用来模拟生物在自然界的遗传环境中由于各种偶然因素引起的基因突变而引入的一种算子。变异算子是把基因串中的某一基因以概率取反运算 (仅仅适用于二进制)。同自然界一样, 每一个基因发生变异的概率很小。变异算子可以对交叉算子起辅助作用以增加算法的全局性, 这是因为交叉算子的作用不能使算法搜索到整个空间, 而变异算子则保证了算法搜索到整个空间中的每一个个体。

(4) 适应度和适应度函数

适应度表示遗传空间中每一个个体对于其环境的适应程度。适应度函数是遗传算法中的一个重要控制参数。类似于优化问题中的目标函数。对于一个给定的优化问题, 如何设计构造适应度函数是一个很关键的问题, 这需要对问题有深入的了解, 我们将通过算法给以详细讨论。

应用遗传算法解决某一问题, 首先要将求解问题的搜索空间映射到遗传空间中的染色体空间中去, 其次, 根据问题和染色体编码构造适应度函数。一般而言, 适应度函数的值越大, 越接近问题的解。再次, 根据问题和可行解的编码结构确定算子, 使得所确定的算子作用在染色体上所得到的子代还在染色体空间中, 并

尽可能的搜索到整个空间。

遗传算法的主要步骤如下，图2.1是它的大致结构：

- 1) 随机地产生一个由确定长度的特征串组成的初始群体；
- 2) 对染色体群体重复执行步骤(i)和(ii)，直到满足停止条件：
 - (i) 计算群体中每个染色体的适应值；
 - (ii) 应用选择、交叉和变异算子产生下一代群体；
- 3) 把群体中适应值最高的染色体直接复制到下一代中，指定任一代中出现的具有最高适应值的染色体为遗传算法的执行结果，这个结果可以表示问题的一个解。

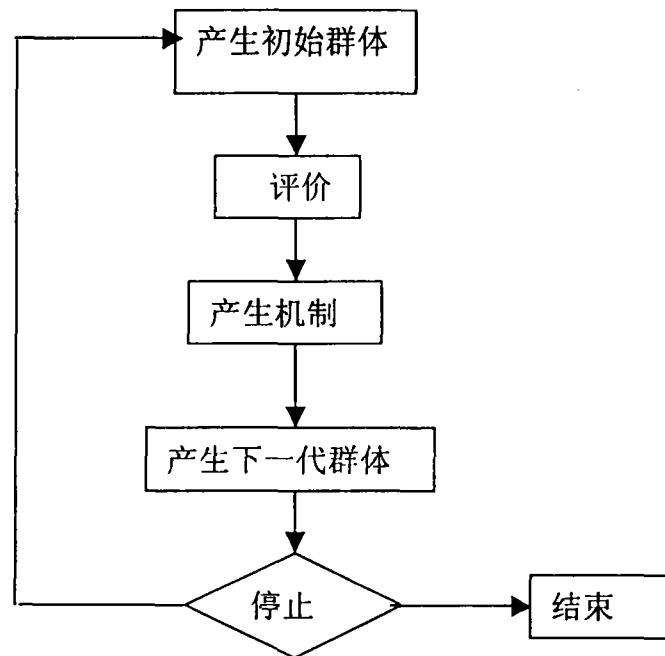


图2.1 遗传算法的步骤

2.4.2 基于遗传算法的多播路由算法

总的来说，基于遗传算法的多播路由算法大体分为两类：基于0、1编码；基于整数编码。

A) 基于0、1编码

这种方法采用二进制编码方式，对所有可能的Steiner节点进行编码。对于多播问题，源节点和目的节点一般是已知的，假定网络的节点数为 n ，目的节点数为 m ，不失一般性，将图中除去源节点和目的节点集合之外的中间节点进行编号为： $0, 1, \dots, n - m - 2$ 。算法对这些节点进行二进制编码，使二进制码的每一位对应一个中间节点。因此二进制码长为 $n - m - 1$ ，形式为： $(q(0), q(1), \dots, q(n - m - 2))$

, 这里 $q(i) \in \{0, 1\}$, $i = 0, 1, \dots, n - m - 2$ 。这个二进制对应的Steiner节点集合 ST 为 $ST = \{v_i \in V \mid q(i) = 1, i = 0, 1, \dots, n - m - 2\}$ 。

对中间节点进行编码是因为, 求解最优Steiner树的关键在于确定包含在最优Steiner树中除目的节点集之外的Steiner节点, 最优Steiner树实质上是包含目的节点集合和Steiner节点的受限最小生成树。

解码操作是关键, 是被研究的最多的部分, 也是此类算法的主要的差异所在。可以说, 此类算法的执行效率以及解的质量的差异主要就是由解码算法决定的。文献^[46]在求解时延受限的多播路由问题时采用的是KMB算法, 其过程如下:

(1) 首先构造包括源节点、目的节点和二进制对应的Steiner节点集合的一个子图 G' ,

这个子图中的每一条边 (i, j) 对应于原图中两个节点 i 和 j 间满足时延约束的一条路径。

(2) 在图 G' 上求一棵满足时延约束的最小生成树 T' ;

(3) 将上面求到的最小生成树 T' 的边还原为原图中的路径; 并删除可能出现的回路。就找到了二进制码对应的时延受限的多播树。在求解两个节点间满足时延约束的路径时用DCUR算法。

文献^[47]采用的解码方式是CKMB^[48], 并在仿真试验中与BSMA算法进行了比较, 取得了较好的效果。此外, 根据解码方式的不同, 还有很多不同的算法相继被提出。但这些算法的编码方式基本都一样。遗传操作时, 采用标准的遗传算子。

B) 基于整数编码

这种方法采用整数编码方式, 对目的节点进行编码。假定目的节点数为 m , 则可将目的节点进行编号为: $0, 1, \dots, m - 1$ 。算法对这些节点进行整数编码, 使整数码的每一位对应一个目的节点。因此二进制码长为 m , 形式为: $(q(0), q(1), \dots, q(m - 1))$, 这里 $q(i) \in \{0, \dots, P(i)\}$, $i = 0, 1, \dots, m - 1$, 其中 $P(i)$ 是从源节点到该目标节点 i 的路径的函数, 其值域为非负整数, 通常是按升序排列的整数。则组成最优Steiner树的路径集合 PT 为 $PT = \{Path_i \mid q(i) = j, i = 0, 1, \dots, m - 1\}$ 。一般说来, $P(i)$ 即为从源节点到达目标节点 i 的路径的数目, 但由于约束条件的存在, $P(i)$ 的值通常会小一些。至于 j , 则是最优染色体的一个基因位。

与上一种编码方式相比, 这种算法通常要作一些预处理工作, 即需要事先计算出备选路径集合, 并映射到非负整数空间, 如果是要解决约束多播问题, 还要将不满足约束条件的路径事先除去。

这类算法并不象前一类的算法那样: 随着遗传操作的进行, 每迭代一次都要

调用相应的解码算法。这样作的好处显而易见，加了遗传操作的速度。而且将遗传操作和预处理操作分离，也使算法更容易实现。

当然，由于预处理后的备选路径集合庞大，可能会使算法收敛较慢。针对这一问题，文献^[49]使用 k 最短路径算法来减小问题的规模。

2.5 随机网络产生模型

我们在研究网络多播路由算法时，一般要分析算法的最坏情况下的性能比，算法的时间复杂度，以及算法的平均性能。因为在很多情况下算法的平均性能要好于最坏情况下的性能，而在实际应用中，往往关心的是算法的平均性能，为此算法的平均性能更为重要。

如何评价算法的平均性能呢？只有通过实验。上面介绍的各种启发式算法都采用了一种共同的实验方法来测试算法的平均性能，这种方法是产生一个随机网络，然后在生成的网络上运行算法，运行多次后取平均值，这样得到的结果一般认为就代表实际平均性能。因此，评估算法性能很重要的一步就是产生仿真用的随机网络；使得产生的网络既具有普遍性，又接近实际网络的特征。在一个特定网络的基础上用某种算法仿真得到的结果，就不具有普遍性，没有说服力。前面讲到的算法都使用相同的随机网络模型，这个随机网络模型是由 Waxman^[50]提出的，由于其具备上述提到的仿真网络应具备的特性，为此被研究多播路由算法的几乎所有的研究者所采用。

本文也采用这个随机网络模型来进行实验，下面介绍 Waxman 随机网络模型的产生方法 Waxman 产生的网络节点是随机分布在矩形网格上的，节点的坐标是一致分布的随机整数，每两个节点之间的距离是他们的欧氏距离，两个节点 u, v 之间以某个概率决定是否将两个节点连接，而这个概率取决于他们之间的距离，节点之间存在边的概率由下面的公式决定

$$P(u, v) = \beta \exp\left(-\frac{d(u, v)}{\alpha L}\right)$$

其中以 $d(u, v)$ 为节点 u 到节点 v 的欧氏距离， L 为节点间最大距离，参数 α 和 β 控制产生网络的特征，其值在 $(0, 1)$ 之间，参数 α 控制随机图中短边与长边， α 增大，图中长边对短边的比值也增大，而 β 控制随机图中的平均点度， β 增大，图的平均点度也增大，调整 α, β 的值就可以得到接近于实际的网络图。但在文献^[50]中并未给出具体的算法步骤，事实上，仅按上边的描述很难控制节点的度数，得到网络往往是一个不十分“均匀”的网络，即，有的节点度数会很大，而其它的节点很小，甚至会有孤立节点的情况。北卡罗来那州立大学的 Salama 和 Reeves

在 Waxman 网络的基础上提出了一个更好的网络生成算法，该算法能更好的控制节点的平均度，并且绝对不会有孤立的节点存在。具体的产生步骤如下：

- 1、计算节点之间的最大距离 L 。
- 2、使节点列表中的第一个节点和另外两个随机选取的节点按概率 $P(u,v)$ 相连。
- 3、对孤立节点，使其和一个随机的非孤立节点按概率 $P(u,v)$ 相连。
- 4、对于度为 1 的节点，使其和一随机点按概率 $P(u,v)$ 相连。
- 5、重复上边的步骤，直到所有的节点的度大于等于 2 为止。
- 6、检查平均节点度数，若小于指定值，则随机选择两个节点按概率 $P(u,v)$ 相连。
- 7、重复第 6 步，直到平均节点度达到指定值为止。

图 2.2, 2.3, 2.4 是 40 节点的网络，它们定制了不同的平均节点度数。

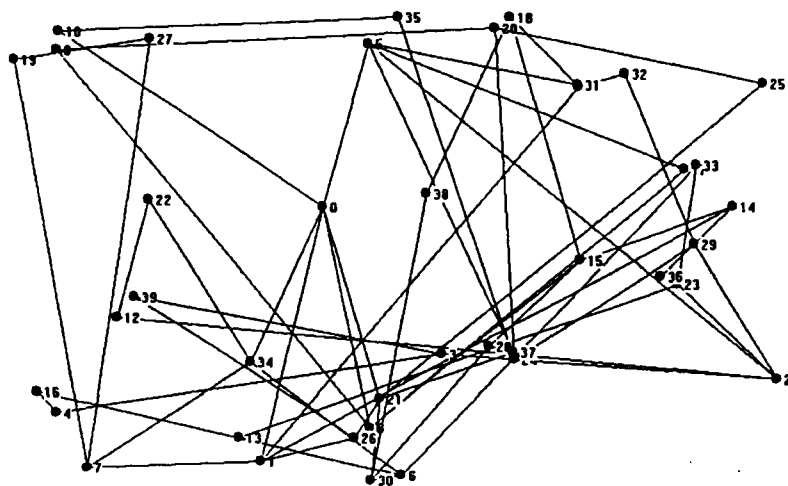


图 2.2 平均节点度为 4

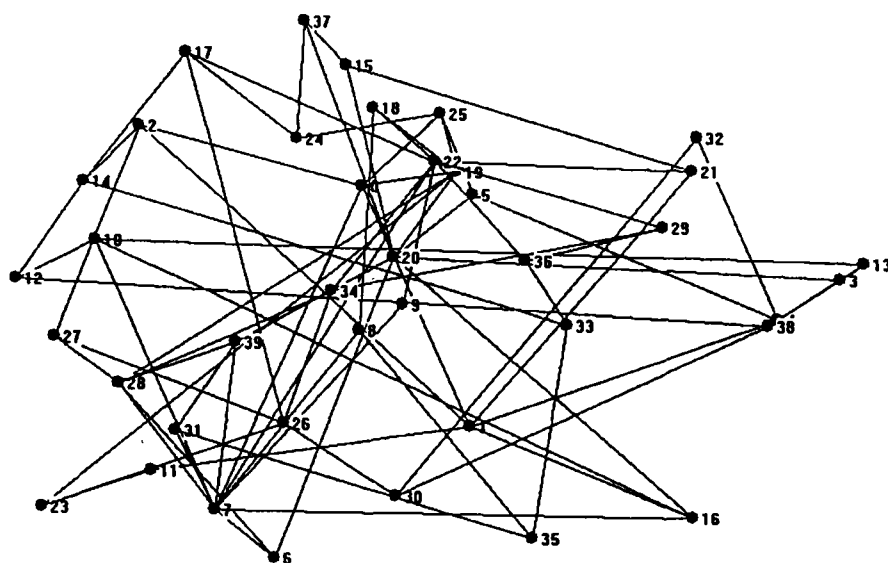


图 2.2 平均节点度为 5

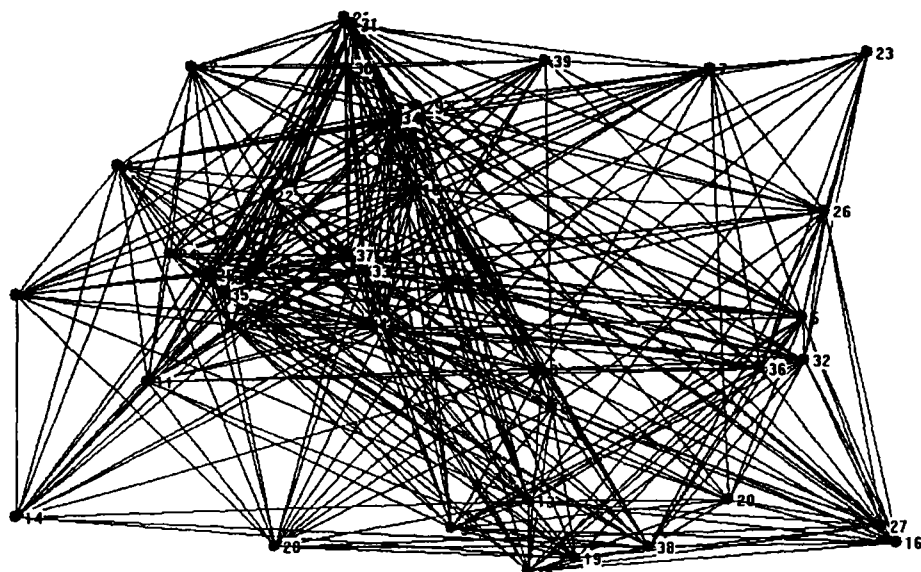


图 2.2 平均节点度为 20

第三章 克隆多播路由算法

3.1 引言

人工免疫系统是模仿自然免疫系统功能的一种智能方法，它实现一种受生物免疫系统启发，通过学习外界物质的自然防御机理的学习技术，提供噪声忍耐、无教师学习、自组织、记忆等进化学习机理，结合了分类器、神经网络和机器推理等系统的一些优点，因此具有提供新颖的解决问题方法的潜力^[51]。其研究成果涉及到控制、数据处理、优化学习和故障诊断等许多领域，已经成为继神经网络、模糊逻辑和进化计算后人工智能的又一研究热点^[52-62]。

本章的人工免疫系统指应用于人工智能领域的免疫机理和由此而形成的方法和理论，即计算机实现的数字化免疫系统模型。其作为智能策略的研究也就是近十几年的事，是人工智能的新兴研究领域。

克隆选择是生物免疫系统理论的重要学说，免疫系统其他可供借鉴的机理还有：记忆学习、反馈机制、多样性遗传机理等。克隆是英文 Clone 一词的单译，意为无性繁殖系，即通过无性繁殖（如细胞丝分裂）可连续传代并形成群体，常用于细胞水平的描述。这一概念已经被广泛的用于计算机编程^[63, 64]、系统控制^[65]、交互式并行仿真等领域^[66]；其学习、记忆等特性同样也引起了人工免疫系统研究者的兴趣^[67, 68]。文献^[67-69]等相继提出了克隆选择算法，但是，相关研究还处在初始阶段，有关算法机理的描述还不多见，相关算子还很少。文献^[70]对克隆策略进行更深入的研究，与其他优化算法进行了比较，并且给出了收敛性证明。根据文献^[70]可知，克隆算法不但能在一定程度上克服早熟，避免算法陷入局部极小值，而且可以提高算法的收敛速度。

本章引入杜海峰等人提出的克隆策略，用于解决时延约束的多播路由问题，即著名的 NP 完全问题，约束 Steiner 树问题。仿真实验表明，同传统的启发式算法相比，该算法具有更快的收敛速度；同时，与基于遗传算法的多播路由算法相比，该算法有效的克服了早熟现象，因此具有更好的全局寻优能力，而且算法稳定、灵活，操作简单。

3.2 免疫克隆选择算法

由于遗传和免疫细胞在增殖中的基因突变，形成了免疫细胞的多样性，这些细胞的不断增殖形成无性繁殖系。细胞的无性繁殖称为克隆。有机体内免疫细胞的多样性能达到这种程度，以至于当每一种抗原侵入机体都能在机体内选择出能

识别和消灭相应抗原的免疫细胞克隆，使之激活、分化和增殖，进行免疫应答以最终清除抗原，这就是克隆选择。但是，克隆—无性繁殖中父代与子代间只有信息的简单复制，而没有不同信息的交流，无法促使进化。因此，需要对克隆后的子代进行进一步处理。

在人工免疫系统中，克隆选择是由亲合度诱导的抗体随机映射，抗体群的状态转移情况可以表示成如下的随机过程：

$$C_s: A(k) \xrightarrow{\text{clone}} A'(k) \xrightarrow{\text{mutation}} A''(k) \xrightarrow{\text{compress}} A(k+1)$$

依据抗体与抗原的亲合度函数亲合度 f^* ，解空间中的一个点 $a_i(k) \in A(k)$ 分裂成了 q_i 个相同的点 $a'_i(k) \in A'(k)$ ，经过变异和选择后获得新的抗体群。在上述过程中，实际上包括了三个步骤，即克隆、变异和压缩选择。

对于二进制编码，抗体 $a \in B^l$ ，其中， $B^l = \{0,1\}^l$ 代表所有长度为 l 的二进制串组成的集合，抗体群 $A = \{a_1, a_2 \dots a_n\}$ 为抗体 a 的 n 元组。克隆算子具体描述如下：

克隆 Θ ：定义

$$\Theta(A) = [\Theta(a_1) \quad \Theta(a_2) \quad \dots \quad \Theta(a_n)]^T \quad (3.2-1)$$

其中： $\Theta(a_i) = I_i \times a_i$ ， $i = 1, 2 \dots n$ ， I_i 为 q_i 维行向量。

$$q_i = g(N, f(a_i)) \quad (3.2-2)$$

一般取：

$$q_i = N * \frac{f(a_i)}{\sum_{j=1}^n f(a_j)} \quad i = 1, 2 \dots n \quad (3.2-3)$$

$N > n$ 是与克隆规模有关的设定值。克隆过后，种群变为：

$$A' = \{A, A'_1, A'_2, \dots, A'_n\} \quad (3.2-4)$$

其中：

$$A'_i = \{a_{i1}, a_{i2}, \dots, a_{iq_i-1}\} \quad a_{ij} = a_i \quad j = 1, 2, \dots, q_i - 1 \quad (3.2-5)$$

免疫基因操作 T_g^c ：免疫基因操作主要包括交叉和变异。根据生物学单、多克隆抗体对信息交换多样性特点的描述，定义仅采用变异的克隆选择算法为单克隆选择算法；交叉和变异都采用的为多克隆选择算法。需要说明的是，这里定义没有完全遵循免疫学上的意义；因为即使对于单克隆算子产生的抗体也不是单一一种抗体，只不过与多克隆算子相比，其更多的保留了父代的特征。免疫学认为，亲和度成熟和抗体多样性的产生主要是依靠抗体的高频变异，而非交叉或重组，因此与一般遗传算法认为交叉是主要算子而变异是背景算子不同，在克隆选择算

法中，更加强调变异的作用。在没有特别指明的情况下，本文的免疫基因操作仅包括变异操作。

变异 T_m^C ：与一般变异不同的是，变异为了保留抗体原始种群的信息，并不作用到 $A \in A'$ ，即：

$$p(T_m^C(a_i) = a'_i) = \begin{cases} p_{ij} > 0 & a_i \in A'_j \\ 0 & a_i \in A \end{cases} \quad (3.2-6)$$

交叉 T_c^C ：同样交叉为了保留抗体原始种群的信息，并不作用到 $A \in A'$ ，即：

$$T_c^C(a_i, a_j) = a'_i \quad a_i \in A'_j, j = 1, 2, \dots, n \quad a_i \in A \quad (3.2-7)$$

克隆选择 T_s^C ： $\forall i = 1, 2, \dots, n$ 若存在变异后抗体 $b = \max\{f(a_j) | j = 2, 3, \dots, q_i - 1\}$ ，使得：

$$f(a_i) < f(b) \quad a_i \in A \quad (3.2-8)$$

则用 b 取代原抗体 a_i ，从而更新抗体群，实现信息交换。

可以看出，克隆的实质是在一代进化中，在候选解的附近，根据亲合度的大小，产生一个变异解的群体，从而扩大了搜索范围；进一步可以认为，克隆是将一个低维空间 (n 维) 的问题转化到更高维 (N 维) 的空间中解决，然后将结果投影到低维空间 (n 维) 中。

免疫克隆选择算法^[93]不是进化算法的简单改进，而是新的人工免疫系统方法。首先，克隆算法的基本思想来源于免疫系统而非自然进化，虽然采用了进化算子，这主要是因为交叉和变异是细胞基因水平上的主要操作，而进化和免疫的生物基础都来自细胞基因的变化。其次，在具体的算法实现上，进化算法更多地强调全局搜索，而忽视局部搜索，但克隆选择算法二者兼顾，并且由于克隆算子地的作用，因而具有更好的种群多样性。再次，进化算法更多地强调个体竞争，较少关注个体间地协作，而以克隆选择算法为代表的人工免疫系统算法则不同；在细胞水平上，由于抗体间的相互协作才有记忆细胞库、疫苗和免疫优势的存在，在算法构造上，不但强调抗体群的适应度函数改变，也关心抗体间的相互作用而导致的多样性改变，提出了抗体—抗体亲合度的概念。最后，在一般遗传算法中，交叉是主要算子，变异是背景算子，但克隆选择算法则恰好相反。

3.3 时延受限的多播路由问题的描述

近年来，随着电视会议、远程教学、视频点播等网络多媒体业务迅速发展，对网络的传输时延，资源的占用提出了更高的要求。多播技术是网络支持多媒体业务的关键技术之一，因此，找到一个有效的多播路由算法是非常必要的。

多播路由是网络中由单一源点到多个指定目的节点的通信形式。确定多播路

由的一般做法是,构造一棵包括源节点和目的节点,且叶子节点为目的节点的多播树,信息沿这棵树确定的路径进行发送时,信息只需在树的分枝处进行复制,这样能够节省网络资源。

多播路由问题属于 NP 完全 (nondeterministic polynomial complete) 问题,针对这一问题,已经提出的无约束多播路由算法有 DNH(Distance Network Heuristic)^[5], SPH(Shortest Path Heuristic)^[7]等算法。针对加了 QoS 限制的多播路由问题,也有很多带约束的多播路由算法,如 BSMA(Bounded Shortest Multicast Algorithm)^[71], CDKS(Constrained Dijkstra Heuristic)^[72]等。这些算法普遍存在的问题是不能很好地平衡多播树的费用与时延限制,算法的复杂度太大。下面首先给出时延受限的多播路由问题的模型。

通常,通信网络可以被表示为一个连通图 $G(V, E)$, V 表示节点(路由器)的集合, E 表示边集合。对于 $\forall u, v \in V$, 分别用 $delay(u, v)$, $cost(u, v)$ 表示弧 (u, v) 上的时延和费用 E 表示边(通信链路)集合。对于 $\forall u, v \in V$, u, v 之间有直接相连的弧 (u, v) , 定义 $delay(u, v)$, $cost(u, v)$ 分别表示弧 (u, v) 上的时延和费用(这里所指的费用是一个广义的概念,可以根据带宽、通信开销、链路距离等因素的函数计算出来)。如果弧 (u, v) 和 (v, u) 具有不同的时延和费用,则 G 是有向图(非对称网络),否则 G 是无向图(对称网络)。

设 S 为多播会话的源节点, D 为信宿节点的集合, $D \cup S \subseteq V$, $|D|$ 表示信宿节点总数。令 $MT = (V_{MT}, E_{MT})$ 表示多播树, V_{MT} 为多播树中节点的集合, E_{MT} 为多播树中边的集合。 $P_{MT}(S, D[i])$ 表示 MT 中从源节点到信宿节点 $D[i]$ 的一条路径。时延约束多播路由问题可以表述为寻找满足 (3.3-1) (3.3-2) 式的一棵生成树:

$$Cost(MT) = \min(\sum_{(u,v) \in E_{MT}} cost(u, v)) \quad (3.3-1)$$

$$\sum_{(u,v) \in P_{MT}(S, D[i])} delay(u, v) \leq \Delta (\Delta \text{ 为时延约束}), i \in 1 \sim |D| \quad (3.3-2)$$

从 (3.3-1), (3.3-2) 式可以看出,多播问题实际上就是一个约束优化问题,即如何求得在 (3.3-2) 式(时延)约束下 (3.3-1) 式(代价)的最小值。

3.4 启发式算法

KPP^[73]是较早提出的一个时延受限多播路由算法,它借鉴了 KMB 算法的思想,首先构造一个只包含源节点和目的节点集合的完全图(complete graph),这个完全图中的每条边对应原图中两个节点之间满足时延约束的费用最短路:然后从完全图中产生一个时延受限生成树;最后将生成树的边用原图中的时延受限最小

费用路径代替，并去掉产生的回路。KPP 算法的复杂度为 $O(\Delta|V|^3)$ 。文献[74]讨论了此算法可能会出现回路，并给出了检查回路和去处回路的算法。从而完善了 KPP 算法。KPP 算法的缺点在于当解存在时，它可能会找不到解。

BSMA(Bounded Shortest Multicast Algorithm)算法^[71]采用先找到问题的可行解，然后对可行解进行改进，使其性能更加接近最优解的方法。BSMA 算法先用 LD 算法生成多播树，然后用费用更小的且满足时延要求的超边（superedge）来代替生成树中费用较大的超边（超边是指一条起始和中止节点都为度大于或等于 2 的节点的一条路径，超边也可以是两个目的节点之间的一条路径，或者是一条度大于 2 的节点和目的节点之间的一条路径）。这个步骤一直反复进行，直到整个生成树的费用不能再减小为止。算法在替换费用高的超边时用到了 k 最短路径算法（ k th-shortest path algorithm）。BSMA 算法求得得多播树得费用较小，缺点是复杂度太高，为 $O(k|V|^3 \log|V|)$ ，不适用求解大型网络得多播路由问题。BSMA 算法可以通过调节参数 k 在多播树费用和计算时间之间进行权衡，降低 k 值，就缩短了计算时间，但同时树的费用会增加。

CDKS(Constrained Dijkstra Heuristic)算法^[72]是在 LD 算法和 BSMA 这类算法之间的一个折中，因为它的复杂度不高，并且得到的多播树的费用较小，仅次于 KPP 算法和 BSMA 算法。该算法先计算一个 LC 树，即用 Dijkstra 算法计算从源节点到各目的节点的费用最短路，然后合并相同链路所构成的树，然后将从源节点到各目的节点的路径中不满足时延要求的路径用 LD 路径替换，从而得到需要的时延受限多播树。这个算法的复杂度为 $O(|V|^2)$ 。

文献^[72]对上面提到的几个算法进行了仿真实验，结果表明 BSMA 算法得到的多播树费用较小，但是它的运行时间太长。CDKS 的总性能最好，它的运行时间较短，费用仅次与 BSMA 和 KPP 算法。

3.5 克隆多播路由算法

3.5.1 符号约定

为了便于说明，先做以下符号约定，令 $clone$ 为克隆算子， mut 为变异算子， sel 为克隆选择算子， f 为亲和度函数。

步骤 1 随机在解空间产生初始抗体群： $X = (x_1, x_2, \dots, x_n)$ ，令 $A = X$ ；

步骤 2 根据以下规则得克隆规模 $\{(k_1, k_2, \dots, k_n); k_i \geq 1, i = 1, \dots, n\}$ 。

$$k_i = \text{Int} \left\lceil N_c * \frac{f(x_i)}{\sum_{j=0}^{n-1} f(x_j)} \right\rceil \quad (3.5-1)$$

N_c 为一大于 n 的整数, $\text{Int}[\bullet]$ 表示上取整数。

步骤 3 克隆(clone)将向量 A 中的每个分量 x_i 分别乘一个 k_i 维的单位向量 I_{k_i} , 即对每个分量 x_i , 分别复制 k_i 份。

$$\begin{aligned} B = \text{clone}(A) &= \{(x_1 I_{k_1}, x_2 I_{k_2}, \dots, x_n I_{k_n}), I_{k_i} = (1, 1, \dots, 1)_{k_i}\} \\ &= \{x_{11}, x_{12}, \dots, x_{1k_1}; x_{21}, x_{22}, \dots, x_{2k_2}; \dots; x_{n1}, x_{n2}, \dots, x_{nk_n}\} \end{aligned} \quad (3.5-2)$$

其中, $x_{ij} = x_i$, B 为经过克隆后所得的抗体群。

步骤 4 变异(mut)变异算子 mut 的选择应根据具体情况确定, 文献[20]中采用的是高斯变异。

$$\begin{aligned} C = \text{mut}(B) &= \{\text{mut}(x_1 I_{k_1}), \text{mut}(x_2 I_{k_2}), \dots, \text{mut}(x_n I_{k_n})\} \\ &= \{x'_{11}, x'_{12}, \dots, x'_{1k_1}; x'_{21}, x'_{22}, \dots, x'_{2k_2}; \dots; x'_{n1}, x'_{n2}, \dots, x'_{nk_n}\} \end{aligned} \quad (3.5-3)$$

步骤 5 克隆选择(sel)

$D = \text{sel}(A, C)$, 且满足

$$\begin{aligned} f(D) = f(\text{sel}(A, C)) &= \{\max(f(x_1), f(x'_{11}), \dots, f(x'_{1k_1})), \\ &\quad \max(f(x_2), f(x'_{21}), \dots, f(x'_{2k_2})), \dots, \max(f(x_n), f(x'_{n1}), \dots, f(x'_{nk_n}))\} \end{aligned} \quad (3.5-4)$$

步骤 6 若满足收敛条件或停机准则, 终止算法; 否则, $A = D$, 转步骤 3。

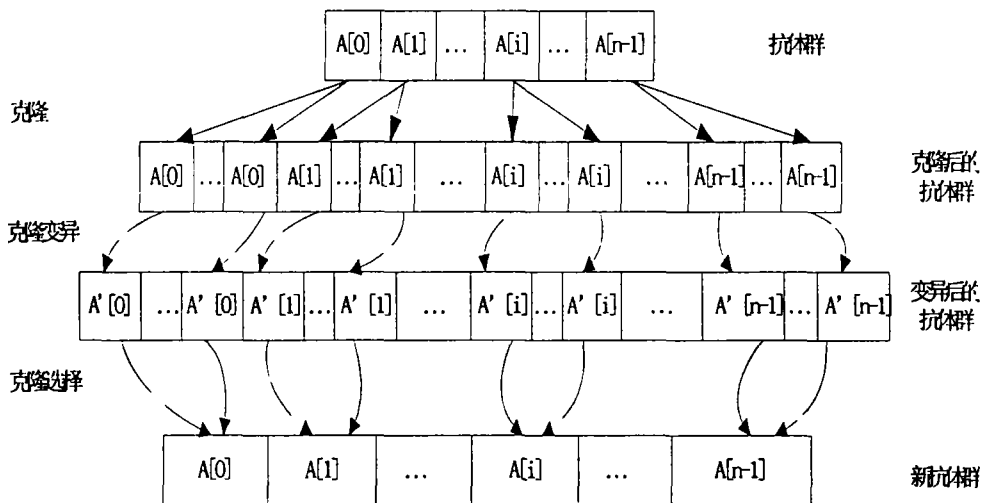


图 3.1 克隆算法的操作过程示意图

可以看出, 克隆的实质是在进化中, 在一代最优解的附近, 根据亲和度的大小, 产生一个变异解的群体从而扩大了搜索范围, 有助于防止进化早熟和搜索陷于局

部极小值。

3.5.2 算法描述

该算法总体分为两步：

- ✓ 求出满足时延条件的备选路径集。
- ✓ 从备选路径集中求出合适的路径组合，构成最优多播树。

(1) 求出备选路径集

利用深度优先搜索找出由源节点 S 到达任一目的节点 $d_i (d_i \in D)$ 的所有满足时延约束的备选路径集 Ω_{d_i} 。具体步骤为：

- 1) 将源节点 S 压入堆栈，并设当前被访问的节点标识 $cur = S$ ；
- 2) 若 $\sum_{(u,v) \in P(S,cur)} delay(u,v) < \Delta$ ，且 cur 节点不在堆栈中，转 3；否则，令其父节点为当前访问节点 cur ，转 5；
- 3) 若 $cur \in D$ ，则记录下来 $P(S,cur)$ 作为一条备选路径。
- 4) 将 cur 压入堆栈。
- 5) 若 cur 有邻接节点没被访问过，访问 cur 的下一邻接节点 $adja$ ，令 $cur = adja$ ，转 2；否则转 6；
- 6) 若 cur 的所有邻接节点都已被访问过，则 cur 退栈，若此时 cur 已是源节点 S ，则转 7；否则，令其父节点为当前访问节点 cur ，转 5；
- 7) 结束算法。

(2) 编码与产生初始抗体群

对于每个多播组端节点 d_j 的备选路径集 Ω_{d_j} ，假设其中有 n_{d_j} 条备选路径，令 $n_i = \max(n_{d_j})$ ，($\forall d_j \in M$)。即 n_i 为具有最多备选路径的目标节点的备选路径数，令集合 $\Omega_i = \{0, 1, 2, \dots, n_i - 1\}$ ，可用 Ω_i 中的元素标识到达任意目标节点的某一条路径。

从集合 Ω_i 中每次任选一数字，共选择 $|D|$ 次，将这 $|D|$ 个数字按选择先后的顺序组成向量，用此向量表示一个初始抗体 x_i 。每个这样的抗体就对应了一棵多播树，而向量中的每个元素 $x_i[r]$ 则对应了到达某一目的节点的一条路径。将上述过程共进行 n 次，可以得到初始抗体群记为 A 。

(3) 计算亲和度

对于那些备选路径数 $n_{d_k} = n_i$ 的目标节点 d_k ，集合 Ω_i 中任意的数都对应到达该目标节点的一条实际路径。此时，定义路径代价为

$$\text{cost}(d_k) = \sum_{(u,v) \in P_T(s,d_k)} c(u,v)$$

对于那些备选路径数 $n_{d_k} < n_i$ 的目标节点 d_k ，当 $j \in [0, n_{d_k} - 1]$ 时， j 对应到达该目标节点的一条实际的路径，其路径代价的定义同上；当 $j \in [n_k - 1, n_i - 1]$ 时， j 并不实际对应一条路径。此时可令 $\text{cost}(d_k) = \text{Max}$ 。可定义亲和度函数为

$$f = \sum \text{Max} - \text{cost}(d_k), d_k \in M.$$

(4) 克隆

对当前抗体群 A 进行克隆操作，既可利用单克隆也可利用多克隆操作。设当前抗体群 $\text{pop}[t]$ 为一 n 维列向量，那么克隆操作可简单表述为：

$$\text{clonepop}_{n \times m} = \{x_1 \times I_m, x_2 \times I_m, \dots, x_n \times I_m\}$$

I_m 即 (3.5-2) 式中 $k_i = k_j = m (i, j \in 1 \sim n)$ 的情况；那么 $\text{clonepop}_{n \times m}$ 即 (3.5-2) 式中的 B 。

(5) 变异

对 $\text{clonepop}_{n \times m}$ 群体进行变异操作，变异概率的选择可以与原始抗体群 n 和克隆规模 m 之积成反比。在文献[70]中使用的是高斯变异，本文为操作简单起见，采用的按位变异，即随机选取一整数 $a \in [0, n_i - 1]$ 按照变异概率取代 $\text{clonepop}_{i \times j}[r]$ ， $(i < n, j < m, r \in [1, D])$ 上原有的值。

(6) 克隆选择

对父代抗体群 A 和变异后的克隆抗体群 $\text{clonepop}_{n \times m}$ 进行克隆选择，组成下一代群体。其选择准则按 (3.5-4) 式进行，即

$$x_i = \begin{cases} \text{clonepop}_{i \times j}, & \text{pop}_{i \times 1} < \text{clonepop}_{i \times j}, j \in [1, m] \\ x_i, & \text{其他} \end{cases}$$

其中 $i \in [1, n]$

计算 A 中每个个体的亲和度，若新一代群体中存在最优个体，则算法中止；否则，转第(4)步。

3.5.3 算法复杂度

本算法由搜索备选路径和利用克隆算法求解最小树两部分组成：设网络节点总数为 n ，时间复杂度为 $O(|D|n^2)$ ；求解最小生成树时，初始编码部分的时间复杂

度为 $O(PopScale \times |D|)$ ，变异部分的复杂度为 $O(N_c \times |D|)$ ，克隆操作的复杂度为 $O(N_c)$ ，因此这一部分总的时间复杂度近似为 $O(K \times N_c \times |D|)$ ， K 为算法的迭代次数。

3.6 仿真实例及结果分析

本文仿真所采用的网络拓扑如图 3.2（文献[46]）所示，图中各边上的树值分别代表费用与时延。源节点和目的节点的取值为 $s=1, D=\{7,11,14,16,18\}$ 。在实验中，最大时延限制分别取为 $\Delta = 9, 25$ 。实验以抗体的亲和度不再变化或到达算法的最大迭代次数为终止条件，最大次数设为 100。本算法取变异概率为 0.5；令 GA 的变异概率为 0.05，交叉概率等于 0.6。两种算法的种群规模都取为网络节点数的 3 倍，克隆策略中的常数 N_c 取为种群规模的 1.5 倍。图 3.3 即为 $\Delta = 25$ 时算出的多播树，其代价为 71，时延为 12。图 3.4 为 $\Delta = 9$ 时的结果，其代价为 92，时延为 8。

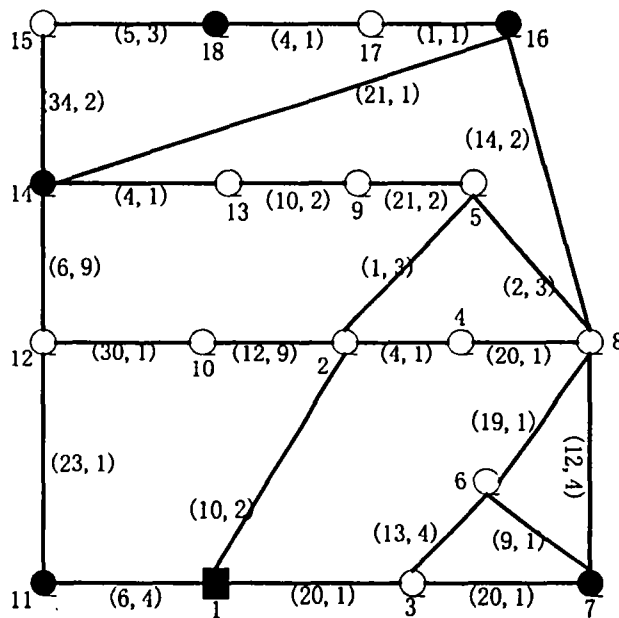


图 3.2 随机网络

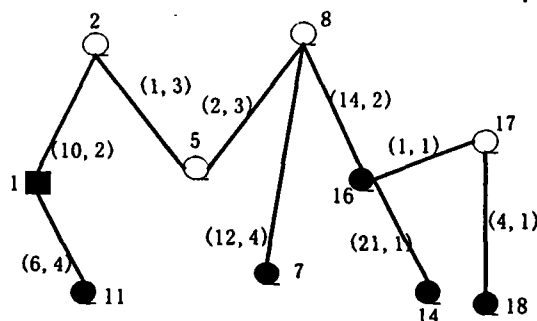
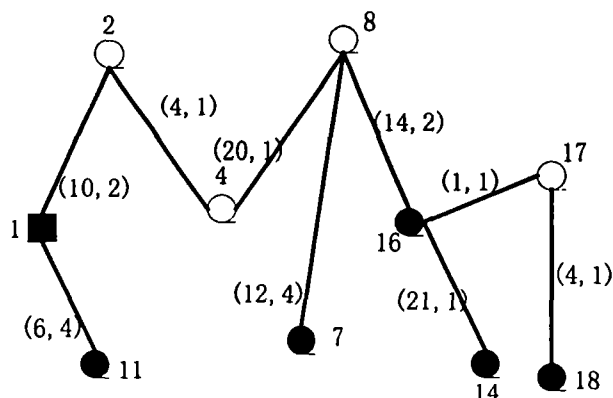


图 3.3 $\Delta = 25$ 时的最优多播树

图 3.4 $\Delta = 9$ 时的最优多播树

在 $\Delta = 25$ 时，100 次实验中，有 36 次都是在 10 代之内即可得到最优解；在 $\Delta = 9$ 时，本算法的平均迭代次数为 4 代，迭代次数大于 10 的仅有 3 次。表 3.1 给出了利用遗传算法^[49]和克隆算法解决图 3.2 所示网络的多播路由问题的收敛分布情况，图 3.5 (a) (b) 是两种算法在不同的时延限制下的多播树费用及时延收敛曲线的比较。可见，当时延限制较小时，本算法的收敛效果与遗传算法相近；当时延变大时，本算法的效果明显优于遗传算法，值得注意的是前者有 11 次只得到了次优解，而本算法在规定的最大迭代次数内不存在得不到最优解的情况。

表 3.1 CS 和 GA 收敛于最优解及次优解的情况

迭代次数	收敛于最优解的次数 ($\Delta = 9$)		收敛于最优解的次数 ($\Delta = 25$)	
	遗传算法 ^[49]	克隆策略	遗传算法	克隆策略
1~10	85	97	7	36
11~30	13	3	35	60
>30	2	0	47	4
收敛于次优解的次数	0	0	11	0

3.7 结论

克隆多播路由算法克服了遗传多播路由算法中存在的早熟现象。但该算法也存在不够稳定的缺点，当时延限制变大时，效果变差，这一点可以从表 3.1 看出。尽管如此，与基于传统遗传算法的多播路由算法相比，基于克隆策略的多播路由算法的优势还是明显的。

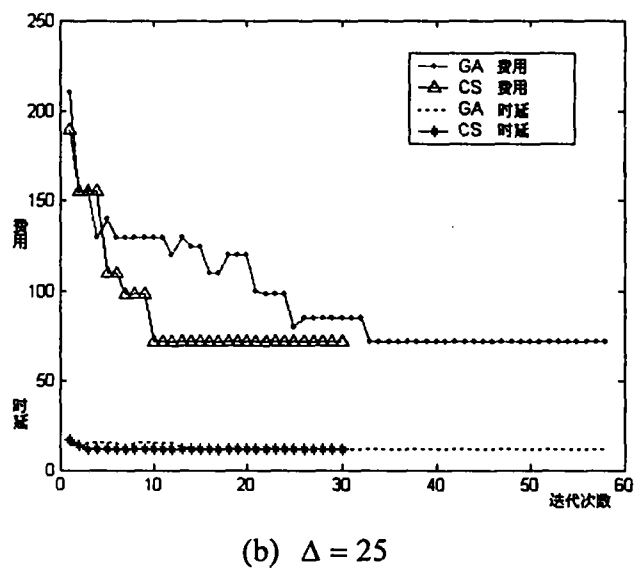
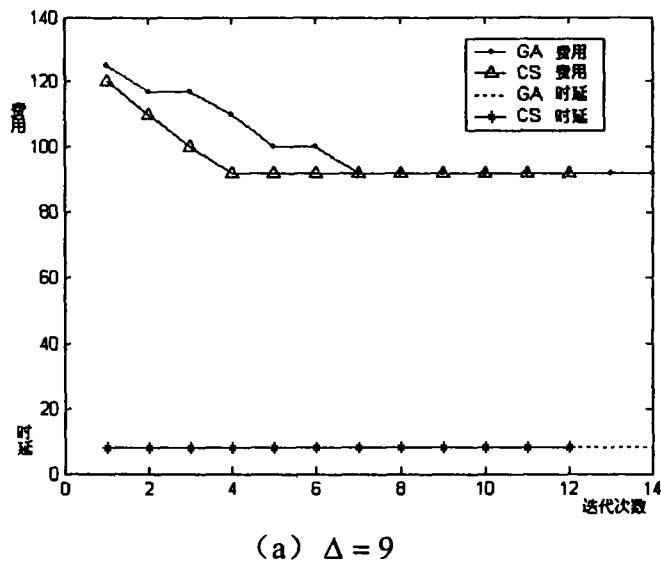


图 3.5 CS 和 GA 搜索多播树费用及时延随迭代次数的变化

第四章 基于免疫算子的克隆多播路由算法

4.1 引言

作为一种 NP 完全问题，多播路由问题即图论中的 Steiner 树^[88]问题多年来一直备受关注。针对这一问题，已经提出了若干启发式算法^{[5][7][71][72]}。这些启发式算法在解决该类问题时往往只利用问题的局部信息，因此通常得不到全局最优的结果。而且算法复杂，实现起来较为困难。

遗传算法(GA)^[84]以其良好的全局寻优能力，近年来越来越多的被用于解决多播路由问题^{[46][49][89]}。并取得了较好的效果。但标准遗传算法只是一种通用优化算法，并未有效的利用与具体问题相关的知识，因此，有时会出现早熟或收敛速度过慢的现象。针对这种情况，王磊等人在文献[90]中提出了免疫算法(Immune Algorithm,IA)，杜海峰等人提出了克隆算法^[70](Clone Algorithm,CA)。

本文提出了免疫克隆多播路由算法(ICBMR, Immune Clone Based Multicast Routing)，是在基于免疫克隆策略的基础上有效地解决了多播路由中的全局优化问题。与传统的遗传算法相比，本算法通过引入与具体问题相关的先验知识来指导搜索过程，加速算法的收敛速度。并且本算法操作简单，易于实现。经过仿真证明，对于一定规模的多播问题，本算法具有较快的收敛速度和较高的准确率。

4.2 免疫算法

4.2.1 产生背景

在生命科学领域中。人们已经对遗传(Heredity)与免疫(Immunity)等自然现象进行了广泛而深入的研究。六十年代，Bagley 和 Rosenberg 等先驱在对这些研究成果进行分析与理解的基础上，借鉴其相关内容和知识，特别是遗传学方面的理论与概念，并将其成功应用于工程科学的某些领域，收到了良好的效果^[75, 76]的至八十年代中期，美国 Michigan 大学的 Holland 教授不仅对以前的学者们提出的遗传概念进行了总结与推广，而且给出了简明，清晰的算法描述，并由此形成目前一般意义上的“遗传算法”(Genetic Algorithm-GA)^[77]。由于遗传算法较以往传统的搜索算法具有使用方便、鲁棒性强便于并行处理等特点，因而广泛应用于组合优化^[78, 79]、结构设计^[80]人工智能^[81]等领域。另一方面，Farmer 和 Bersini 等人也先后在不同时期、不同程度地涉及到了有关免疫的概念^[82, 83]。

众所周知，遗传算法是一种具有“生成+检测”(generate-and-test)的迭代过程的搜索算法^[84]。从理论上分析，迭代过程中，在保留上一代最佳个体的前提下，遗传算法是全局收敛的^[85]。然而，在对算法的实施过程中不难发现两个主要遗传算子都是在一定发生概率的条件下。随机地、没有指导地迭代搜索，因此它们在为群体中的个体提供了进化机会的同时，也无可避免地产生了退化的可能。在某些情况下这种退化现象还相当明显。另一方面，每一个待求的实际问题都会有自身一些基本的、显而易见的特征信息或知识。然而遗传算法的交叉和变异算子却相对固定。在求解问题时，可变的灵活程度较小。这无疑对算法的通用性是有益的，但却忽视了问题的特征信息时求解问题的辅助作用，特别是在求解一些复杂问题时，这种“忽视”所带来的损失往往就比较明显了。实践也表明，仅仅使用遗传算法或者以其为代表的进化算法，在模仿人类智能处理事物的能力方面还远远不足。还必须更加深层次地挖掘与利用人类的智能资源。从这一点讲，学习生物智能，开发，进而利用生物智能是进化算法乃至智能计算的一个永恒的话题。

基于这一思想，王磊将免疫概念及其理论应用于遗传算法。在保留原算法优良特性的前提下，力图有选择、有目的地利用待求问题中的一些特征信息或知识来抑制其优化过程中出现的退化现象，这种算法称为免疫算法(Immune Algorithm-IA)。这种算法较好地解决了遗传算法中的退化问题。

4.2.2 基本概念

为了便于说明，首先对免疫算法中的基本概念进行解释。

定义 4.1 染色体

染色体(Chromosome)，表示待求问题的一种数据结构，如数组或位串等。

定义 4.2 基因

基因(Gene)，是指构成染色体的最基本的数据单位。例如，如果表示染色体的数据结构为数组，则基因一般是指其中的元素；如果染色体的数据结构为位串，则基因一般即指其中的位。

定义 4.3 个体

个体(Individual)，具有某类染色体结构的一种特例。

定义 4.4 抗原

抗原(Antigen)，是指所有可能错误的基因，即非最佳个体的基因。比如，同样是上述类型的染色体，如果最佳个体为第一种结构，则所有与该结构上对应位的基因上不同的基因均可视为抗原。

定义 4.5 疫苗

疫苗 (Vaccine), 是指根据进化环境或待求问题的先验知识, 所得到的对最佳个体基因的估计。

定义 4.6 抗体

抗体 (Antibody), 是指根据疫苗修正某个个体的基因所得到的新个体。其中, 根据疫苗修正个体基因的过程即为接种疫苗 (Vaccination), 其目的是消除抗原再新个体产生时所带来的负面影响。

4.2.3 免疫概念

免疫概念的提出是受生物自然科学的启发^[86]。具体的实现过程中, 在原有标准遗传算法的框架基础上引入了一个新的算子, 即免疫算子 (Immune Operator)。同生命科学中的免疫理论类似, 免疫算子也有两种类型: 全免疫 (Full Immunity) 和目标免疫 (Target Immunity), 二者分别对应于生命科学中的非特异性免疫和特异性免疫。其中, 全免疫是指群体中每个个体在遗传算子作用后, 对其每一环节都进行一次免疫操作的免疫类型; 目标免疫则是指在进行了遗传操作后, 经过一定判断, 个体仅在作用点处发生免疫反应的一种类型。前者主要应用于个体进化的初始阶段, 而在进化过程中基本上不发生作用, 否则将很有可能产生通常意义上所说如“同化现象” (Assimilative Phenomenon); 后者一般而言将伴随群体进化的全部过程, 它是免疫操作的一个基本算子。实际的操作过程中, 首先对所求解的问题 (这里视为抗原) 进行具体分析, 从中提取出的最基本的特征信息 (即疫苗, Vaccine) 其次, 对此特征信息进行处理, 以将其转化为求解问题的一种方案 (根据该方案而得到的各种解的集合统称为基于上述疫苗所产生的抗体) 最后, 将此方案以适当的形式转化成免疫算子以实施具体的操作。这里需要说明的是, 待求问题的特征信息往往不上一个, 也就是说针对某一特定的抗原所能提取出的疫苗也可能不是一种, 那么在接种疫苗过程中可以随机地选取一种疫苗进行注射, 也可以将多个疫苗按照一定的逻辑关系进行组合后再予以注射。

综上所述, 免疫思想主要是在合理提取疫苗的基础上, 通过接种疫苗和免疫选择两个操作步骤来完成的前者是为了提高适应度, 后者则为了防止群体的退化。具体而言, 它们分别是:

接种疫苗: 设个体 x , 给其接种疫苗是指按照先验知识来修改 x 的某些基因位上的基因, 使所得个体以较大的概率具有更高的适应度。首先考虑以下两种特殊情况: 其一, 若个体 y 的每一基因位上的信息都是错误的, 即每一位码都与最佳个体不同, 则任一个体 x , x 转移为 y 的概率为 0; 其二, 若个体 x 的每个基因位都是正确的, 即 x 已经是最佳个体, 则 x 以概率 1 转移为 x 。除此之外, 设有群

体 $c = (x_1, x_2, \dots, x_{n_0})$ ，对 c 接种疫苗是指在 c 中按比例 α 随机抽取 $n_\alpha = \alpha n (0 < \alpha \leq 1)$ 个个体而进行的操作。疫苗是从对问题的先验知识中提炼出来的，它所包含的信息量及其准确性对算法的性能起着重要的作用。

免疫选择：这一操作分两步完成。第一步是免疫检测，即对接种了疫苗的个体进行检测，若其适应度仍不如父代，说明在交叉、编译的过程中出现了严重的退化现象。这时该个体将被父代中所对应的个体所取代；如果子代适应度优于父代则进行第二步处理。第二步是退火选择^[87]，即在目前的子代群体 $Ek = (x_1, \dots, x_{n_0})$ 中以概率：

$$P(x_i) = e^{f(x_i)/T_k} / \sum_{i=1}^{n_0} e^{f(x_i)/T_k}$$

选择个体 x_i 进入新的父代群体，其中 $f(x_i)$ 为个体 x_i 的适应度， $\{T_k\}$ 是趋近于 0 的温度控制序列。

4.2.4 算法描述

1. 随机产生初始父代种群 A_1 。
2. 根据先验知识抽取疫苗。
3. 若当前群体中包含最佳个体，则算法停止运行并输出结果：否则继续。
4. 对于目前的第 k 代父本种群 A_k 进行交叉操作，得到种群 B_k 。
5. 对 B_k 进行变异操作，得到种群 C_k 。
6. 对 C_k 进行接种疫苗操作，得到种群 D_k 。
7. 对 D_k 进行免疫选择操作，得到新一代父本 A_k 。转至 3。

该算法的运行流程可对照图 4.1：

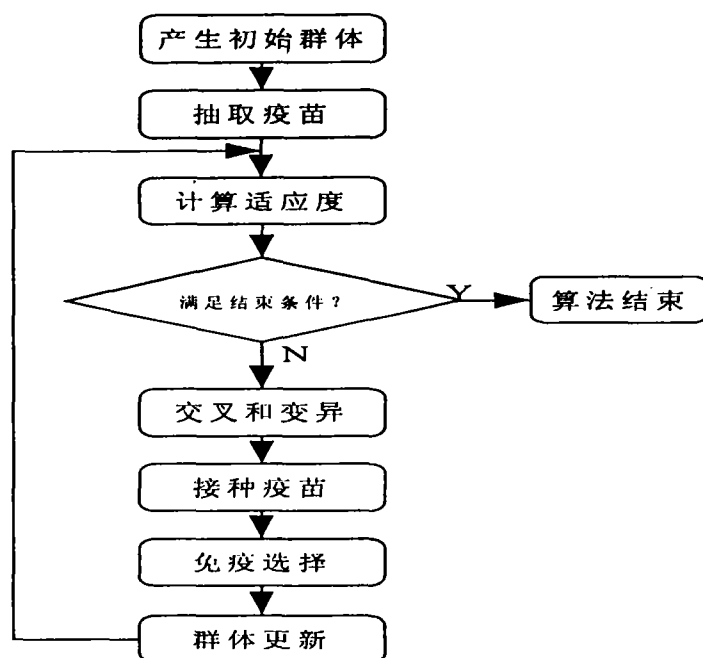


图 4.1 免疫算法操作步骤

4.2.5 免疫算子的机理与构造

免疫算子是由接种疫苗和免疫选择两部分操作构成的，其中疫苗指的是依据人们对待求问题所具备的或多或少的先验知识，从中提取出的一种基本的特征信息；抗体是指根据这种特征信息而得出的一类解。前者可以看作是对待求的最佳个体所能匹配模式（Schema）的一种估计；后者则是对这种模式进行匹配而形成的样本。不难发现，疫苗的正确选择对算法的运行效率具有十分重要的意义。它如同通用遗传算法中的编码一样，是免疫操作得以有效地发挥作用的基础与保障。但是需要说明的是：选取疫苗的优劣，生成抗体的好坏，只会严重影响到免疫算子中接种疫苗作用的发挥，不至于涉及到算法的收敛性。

免疫算法的应用对象主要是针对一些难度随规模扩大而迅速增大的问题，如 NP 问题等。这类问题的特点是在规模较小的，问题一般易于求解或者说易于发现其局部条件下的求解规律。针对这一类问题选取疫苗时（可以推广到一般问题），既可以根据问题的特征信息来制作免疫疫苗，也可以在具体分析的基础上，考虑降低原问题的规模，增设一些局部条件来简化问题。这种简化后的问题求解规律就可做为选取疫苗的一种途径。不过在实际的选取过程中，应考虑到：一方面，原问题局域化处理越彻底，局部条件下的求解规律就越明显，这时虽然易于获取疫苗，但寻找所有这种疫苗的计算量会显著增加；另一方面，每一个疫苗都是利

用某一局部信息来探求全局最优解，即估计该解在某一分量上的模式，所以没有必要对每个疫苗做到精确无误。因此一般可以根据对原问题局域化处理的具体情况，选用目前通用的一些迭代优化算法来提取疫苗。

4.3 基于免疫算子的克隆多播路由算法

4.3.1 符号说明

$Graphic(V, E)$: 连通简单图，既可是有向的，又可是无向的，表示通信网络。

V : 网络中节点的集合。

E : 表示通信链路集合。对于 $\forall(u, v) \in E$ ，定义非负值 $delay(u, v)$ ，

$delay(u, v)$: 值非负， (u, v) 上的时延， $\forall(u, v) \in E$ 。

$cost(u, v)$: 值非负， (u, v) 上的费用， $\forall(u, v) \in E$ 。

$Source$: 多播会话的源节点

$DestSet$: 信宿节点的集合， $DestSet \cup Source \subseteq V$ 。

$|DestSet|$: 表示信宿节点总数。

MT : 多播树。 $(V_{MT} \subseteq V, E_{MT} \subseteq E)$ 。

V_{MT} : 多播树中的节点， $V_{MT} \subseteq V$ 。

E_{MT} : 多播树中的边， $E_{MT} \subseteq E$ 。

4.3.2 算法描述

本文提出算法结合了免疫算法和克隆算法的优势。算法描述如下：

1 求解备选路径集

求出满足 (3.3-2) 式的路径组成备选路径集合。

2 克隆多播路由算法

将上一步计算出的备选路径放在名为 $DestSet$ 的动态数组中，数组的长度即为

请求多播会话的组成员数 $|DestSet|$ ，数组元素的数据结构如下：

```

DestSet[i]{
    ID;//请求多播会话的节点的标识
    PathSet; //从目的到该源节点的备选路径集合
    PathCounts; //到该源节点的备选路径总数
}

```

(1) 编码机制

用抗体 *Antigen* 表示一棵多播树，其编码长度为 $|DestSet|$ ，每个码位 *Antigen*[*i*] 的取值为 $0 \sim (PathCounts - 1)$ 之间的任一整数，对应 *DestSet*[*i*].*PathSet* 中的一条路径。

(2) 亲和度函数的建立

亲和度函数可以定义如下：

$$\Phi(Antigen) = 1 / COST(Antigen) \quad (4.3 - 1)$$

(3) 产生初始抗体群

抗体群记为 *AntigenSet*。其中的每个元素 *AntigenSet*[*i*] ($i \in 0 \sim PopScale - 1$ ，*PopScale* 是抗体群规模) 即 *Antigen* 采用随机产生的方式，具体方法为：将一个随机整数 $RAND \in [0, PathCounts - 1]$ 赋给 *Antigen*[*j*]，($j \in 0 \sim |DestSet| - 1$)。

(4) 抽取疫苗

由 Prim 和 Kruskal 算法在产生最小生成树的过程可以看出，他们采用的都是贪婪算法^[50]的思想，即逐次将与生成树之间有最短边的节点加入树中。可以断定，组成最优多播树的路径必然由到达某些目的结点的最短路径和被共享的较多的次短路径组成。因此选择满足时延条件的最短路径树 (SPT) 作为疫苗，这个疫苗利用了问题的部分先验知识。

(5) 克隆

对抗体群 *AntigenSet* 中的每一个抗体按如下规模克隆 q_i 到新的抗体群中，记为

CloneAntis :

$$q_i = Int \left[N_c * \frac{\Phi(AntigenSet[i])}{\sum_{j=0}^{PopScale-1} \Phi(AntigenSet[j])} \right] \quad (4.3 - 2)$$

N_c 为一大于 *PopScale* 的整数，本文取为 *PopScale* 的 1.5 倍。 $Int[\bullet]$ 表示上取整数。

(6) 接种疫苗

按概率 P_i 抽出新抗体群 $CloneAntis$ 亲和度最小的一些抗体，用算法第 (4) 步得到疫苗，即最短路径树代替。

(7) 交叉和变异

交叉和变异操作只对新抗体群 $CloneAntis$ 进行而不对 $AntigenSet$ 进行。采用单点“与或”交叉操作，即两个抗体的交叉点后的各码位若码值相等则保留，若不等则分别用各自备选路径集中代价较小的任一路径代替；变异时按变异概率对每个抗体 $Antigen$ 的每个码位赋 $0 \sim (PathCounts-1)$ 之间的随机整数，与进化算法相比，克隆策略的变异概率要大一些。

(8) 克隆选择

选择过程根据公式 (4.3-3) 进行。

$$P_s = \begin{cases} 1 & \Phi(A[i]) < \Phi(B) \\ \exp(-\frac{\Phi(A[i]) - \Phi(B)}{a}) & \begin{aligned} &\Phi(A[i]) \geq \Phi(B) \\ &\text{, 且 } A[i] \text{ 不是目前种群的最优抗体} \end{aligned} \\ 0 & \begin{aligned} &\Phi(A[i]) \geq \Phi(B) \\ &\text{, 且 } A[i] \text{ 是目前种群的最优抗体} \end{aligned} \end{cases} \quad (4.3-3)$$

4.3.3 算法复杂度分析

假设网络节点总数为 n ，目标节点数为 $DestCount$ 。本文的算法复杂度分为搜索备选路径的时间复杂度和利用免疫克隆算法求解最优多播树的时间复杂度两部分。搜索备选路径的时间复杂度为 $O(DestCount \times n^2)$ ；用免疫克隆算法求最优多播树时，接种疫苗部分的算法复杂度为 $O(S \times N_c)$ ，克隆操作的时间复杂度为 $O(S \times N_c)$ ，因此这一部分总的时间复杂度近似为 $O(S \times N_c)$ ， S 为算法的迭代次数。由此，本文提出算法总的时间复杂度为 $O(DestCount \times n^2) + O(S \times N_c)$ 。

4.4 仿真实验

本实验采用与实验 3.6 相同的网络拓扑结构。并且取相同的源及目标节点的集合。

分别利用本文的算法和遗传算法^[49]来求解最优多播树。为了便于比较，两种

算法的变异概率均取为 0.07，交叉概率取为 0.6，种群规模都取为网络节点数的 3 倍，免疫克隆算法中的常数 N_c 取为种群规模的 1.5 倍，算法以达到迭代次数上限为结束条件，该上限取为 100。图 4.2 (a) (b) 是两种算法在不同的时延限制下的得到的多播树代价的收敛曲线的比较，该曲线结果由 100 次试验的平均值所得。可以看出，与传统的遗传算法相比，本文提出的算法有更快的收敛速度，并且针对不同的时延限制，本文提出的算法所表现的性能也更为稳定。表 4.1 显示了两种算法收敛于最优解和次优解时，迭代次数的分布情况。

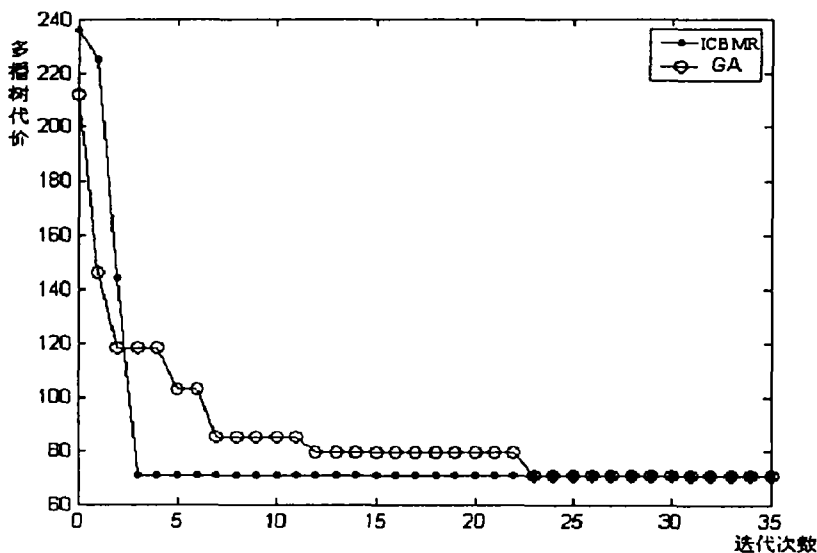
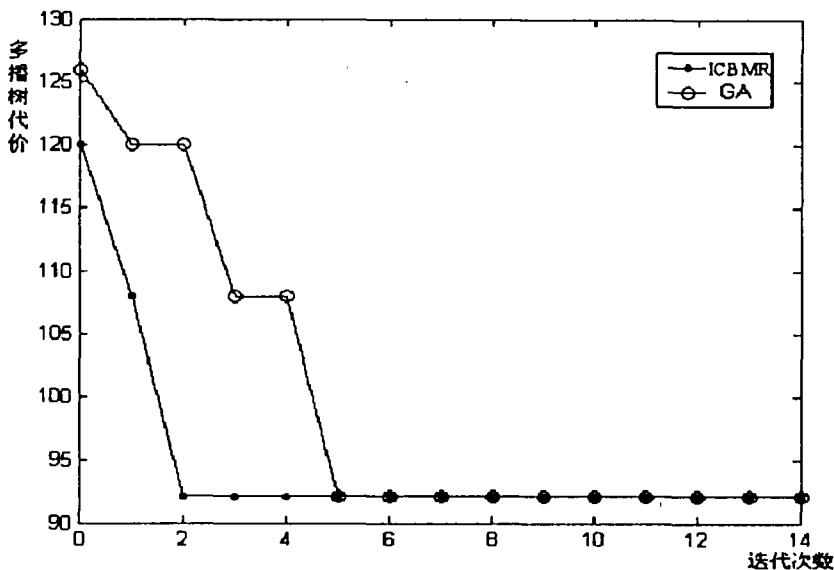
(a) $\Delta = 25$ (b) $\Delta = 9$

图 4.2 ICBMR 和 GA 搜索多播树代价随迭代次数的变化

表 4.1 ICBMR 和 GA 收敛于最优解及次优解的情况

迭代次数	收敛于最优解的次数 ($\Delta = 9$)		收敛于最优解的次数 ($\Delta = 25$)	
	遗传算法 ^[49]	免疫克隆算法	遗传算法	免疫克隆算法
1~10	93	100	10	100
11~30	5	0	41	0
>30	1	0	39	0
收敛于次优解的次数	1	0	10	0

从图 4.3~4.5 可以看出, 本文算法所得的多播树的费用明显小于其他两种算法。图 4.7 是 GA 和本文算法作用于图 4.6 所示网络时的迭代情况的比较, 算法开始时, GA 所得的多播树费用较小, 迭代到 40 代以后, GA 与本文算法的比值就逐渐变大, 说明 GA 陷入了早熟, 而本文算法则找到了代价更小的多播树。

4.5 结论

本文通过引入免疫算子, 有目的地利用待求问题中的一些特征信息或知识, 从而有效的抑制了利用遗传算法解决优化问题中的退化现象, 并结合克隆算法解决多播路由问题, 取得了较快的速度和较好的结果。而且算法实现简单, 可扩展性良好, 有较好的应用前景。

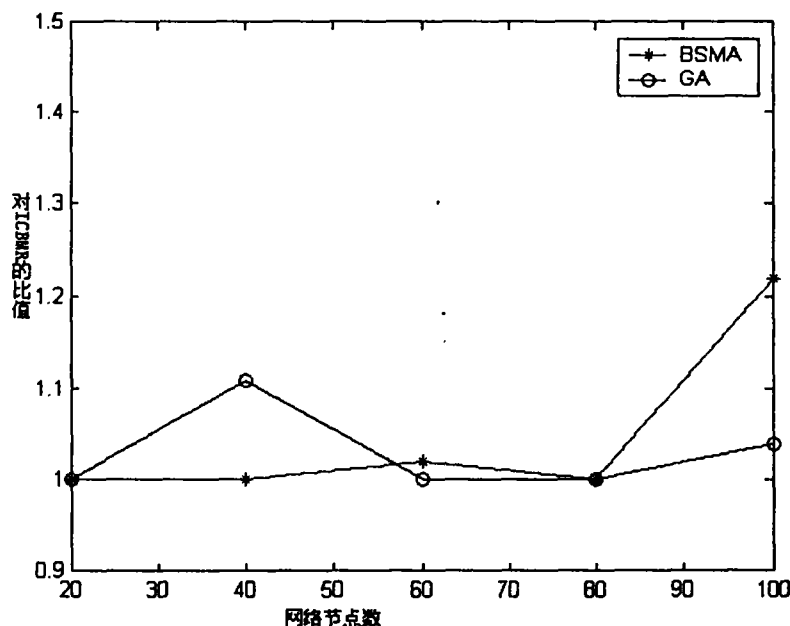


图 4.3 目标节点占网络节点总数 5% 时, GA、BSMA 所得多播树的费用和 ICBMR 的比值

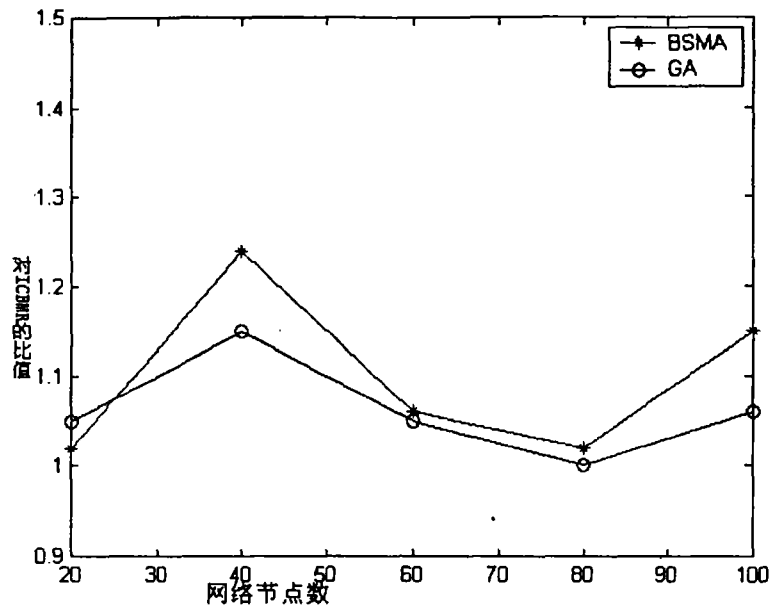


图 4.4 目标节点占网络节点总数 15% 时, GA、BSMA 所得多播树的费用和 ICBMR 的比值

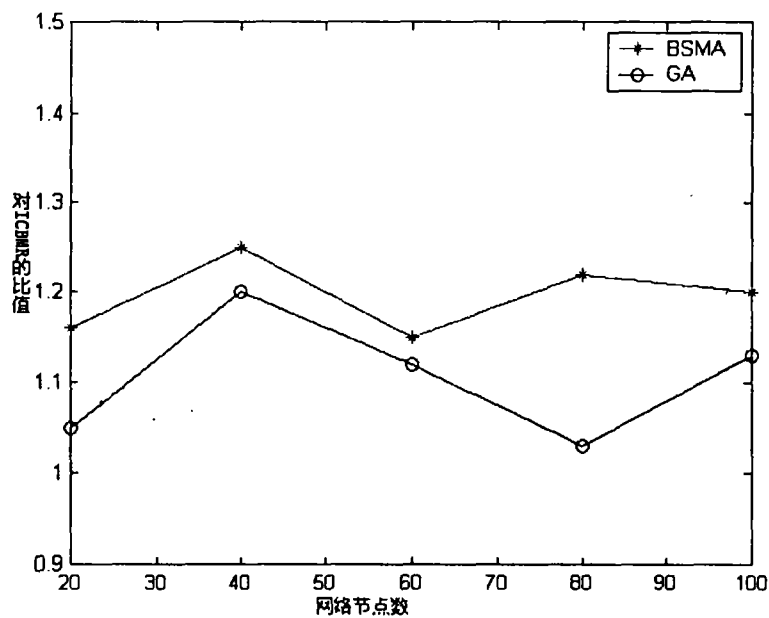


图 4.5 目标节点占网络节点总数 30% 时, GA、BSMA 所得多播树的费用和 ICBMR 的比值

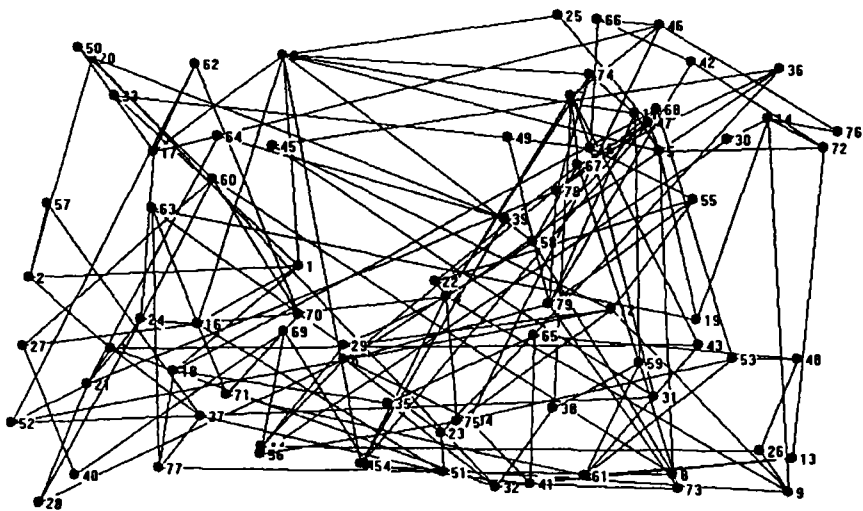


图 4.6 80 节点的随机网络

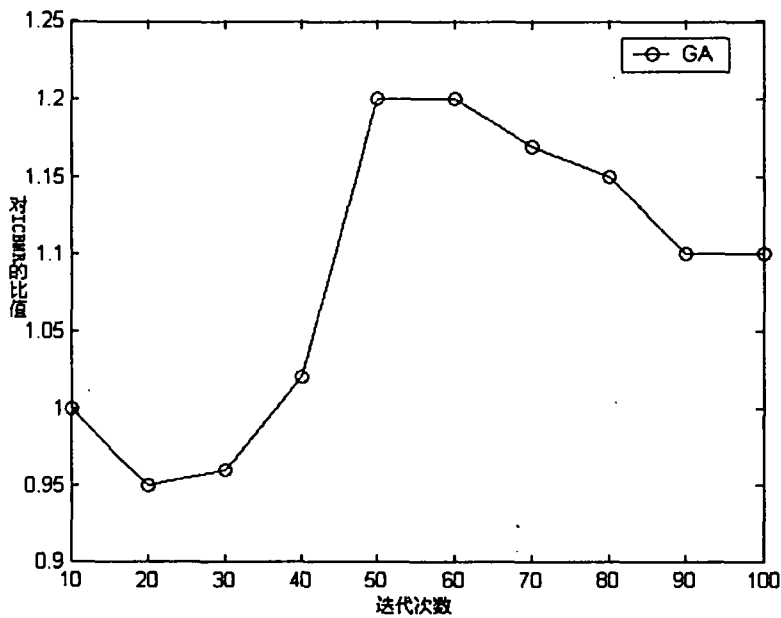


图 4.7 目标节点占网络节点总数 30% 时，GA 所得多播树的费用和 ICBMR 的比值

第五章 参数可调的克隆多播路由算法

5.1 引言

在过去几年里,已经提出了很多启发式多播路由算法^{[71][73][91][92]},其中 BSMA^[71]算法被认为是最好的受限多播路由算法。然而,过长的计算时间限制了其应用,事实上,这也是大多数确定性算法普遍存在的问题,除此以外,这些算法还存在复杂度太大,实现复杂,在有限的时间内都不能得到费用较小的多播树的缺点^[72]。

因此,人们将目光转向了遗传算法^[84]。从多数仿真看来,遗传算法取得了比确定性算法更好的结果,但其易“早熟”的特性也影响了算法性能的进一步提高。为此,本文提出克隆多播路由算法,该算法有效地克服了早熟现象。而且通过引入一个可调因子,缩小了算法的无效搜索空间,加快了收敛速度。仿真表明,该算法是有效可行的,与 BSMA 算法和进化算法解决同类问题相比具有更令人满意的结果。

5.2 参数可调的克隆多播路由算法

5.2.1 算法描述

1、求解备选路径集

假设 *Graphic* 是一个总节点数为 n 的完全图,即网络中每个节点的度都为 $n-1$,则从网络中一节点出发到达另一指定节点之间的可行路径共有:

$$PathCount = 1 + P_{n-2}^1 + P_{n-2}^2 + P_{n-2}^3 + \cdots + P_{n-2}^{n-2} > n! \text{ 条}$$

若在这样一个网络上构建多播树,可供选择的组合数约有:

$$\underbrace{PathCount \times PathCount \times \cdots \times PathCount}_{DestCounts}$$

对于一个无约束的全连通网络来说,其备选路径集十分庞大,但是加上时延约束后便会使部分原来连通的路径因为不满足时延条件而变的不再连通,问题的规模因而被缩小了。时延约束越紧,则备选路径集也越小。以图 3.2 所示的随机网络为例,当时延约束 $\Delta = 9$ 时,到达各目标节点的平均路径数为 3 条,当 $\Delta = 12$ 时,平均路径 6 条,当 $\Delta = 25$ 时,平均路径为 30 条。

从上文的分析可以看出,当时延约束放宽时,备选路径的数目很多。而真正

构成最优多播树的路径往往是备选路径集合中代价较小的几条。例如图 3.2 中，构成最优的多播树的 5 条路径都是备选路径集中代价最小和次小的路径。这个规律可以从 Prim 和 Kruskal 算法中可以看出，这两种算法采用的都是贪婪算法^[50]的思想，逐次将与生成树之间有最短边的节点加入树中。因此，在一个最小生成树中，从源节点到某个目标节点的路径代价一定在从源节点到这个目标节点的所有路径中也是较小的，但不一定是最小的。因此，有必要减少备选路径的数目。

基于上述分析，本文的做法，是先求出从目的节点到源节点的所有满足时延条件的路径，按代价升序排列，然后通过引入一个动态可调参数，在每次路由算法开始时，只选择备选路径集中的前 $k\%$ 进行组合，并且，在路由算法进行过程中， k 值也可以随时调节，并且基本不会增加额外的负担。

本文选择备选路径的算法的伪代码如图 5.1。由于算法每步扩展的都是未访问节点并对时延进行检测，因此选出的路径都满足时延限制并且不会出现环路。

<pre> 1. recurSearRout(int ID) { 2. //先将当前就要访问的的节点压入栈中 3. CurrentLinkStack.push_back(ID); 4. //设置已访问标志 5. setVisitSig(ID,true); 6. //如果分组已经到达Source 7. if(ID == SourcID){ 8. 添加这一条到达源节点的路径; 9. //ID节点退栈 10. CurrentLinkStack.pop_back(); 11. return; 12. } 13. //如果当前节点不是目标节点, 14. //对其邻居如下操作 </pre>	<pre> 15. for(int i = 0; i < ID.getDegree; i++){ 16. tempID = ID.getNeighbor(i); 17. if(加入tempID节点后的时延 18. <= Δ && tempID没有在栈中){ 19. //递归调用 20. recurSearRout(tempID); 21. } 22. } 23. //当前节点退栈 24. CurrentLinkStack.pop_back(); 25. //恢复未访问标志 26. setVisitSig(ID,false); 27. return; 28. } </pre>
---	---

图 5.1 备选路径算法

为了加快算法的收敛速度，必须缩小搜索空间，即备选路径集合。根据最小生成树^[50]算法可以假定，组成最优多播树的路径必然包含到达某些目的结点的最短路径。因此本文的做法是只选择备选路径集中代价最小的若干路径组成新的“备选路径集”。在路由算法的具体实施过程中，可以通过引入可调参数 k (k 的含义参见 5.2.2 节中的“编码机制”) 来扩大或缩小“备选路径集”，这样做基本不会增加算法的额外负担。

2、参数可调的克隆多播路由算法

信源节点备选路径信息放在名为 D 的数组中，数组的长度即为请求多播会话

的组成员数 $|D|$ ，数组元素的数据结构如下：

```

D[i]{
    ID=i; //请求多播会话的节点的标识
    P; //从目的到该源节点的路径集合
    |P|; //到该源节点的路径总数
}

```

(1) 编码机制

目标节点的个数 $|D|$ 就是抗体 *Antigen* 的编码长度，每个码位 *Antigen*[*i*] 对应各自 *D*[*i*]，抗体 *Antigen* 就对应一棵多播树。源节点到每个目标节点的备选路径数量的多少决定了搜索空间的大小。为了获得更为有效和相对较小的搜索空间，本文引入了动态可调参数 $k(k \in (0, 100))$ ，令抗体 *Antigen* 每个码位 *Antigen*[*i*] 的取值范围为 $0 \sim |P| \times k\% - 1$ 之间的任一整数，该码值的含义是路径集合 *D*[*i*].*P* 中代价最小的 $\text{Int} \lceil |P| \times k\% \rceil$ 条路径中的一条。由此编码机制形成的搜索空间是动态可变的。本文中 k 值是按如下方法进行调节：

1. 先令 k 取一个较小的值 k_0 ，步长 $d_k = d_0 > 0$ ，且 $k_0\%, d_0\% \ll \alpha$ ；
2. 进行克隆多播路由算法操作。
3. 如果最优亲和度连续 n 次 (n 的取值应小于作为算法结束的条件迭代次数) 不变，转 4；否则转 2。
4. 令 $k = k + d_k$ ，若 $k\% \geq \alpha$ ，令 $d_k = -d_0$ ， $k = \alpha \times 100 + d_k$ 。
5. 如果 $d_k > 0$ ，令 $d_k = d_k + d_0$ 。
6. 若 $k \leq k_0$ ；停止算法；否则转 2。

其中 α 的取值为 0.4~0.5 之间的一个浮点数 (α 的取值为一经验值)， k_0 的值视 $|P|$ 的大小而定，当 $|P|$ 较大时， k_0 可以取的相应小一些。在这样的取值下可缩小搜索空间，使算法收敛速度加快。

(2) 亲和度函数的建立

亲和度函数可以定义如下：

$$\Phi(\text{Antigen}) = 1 / \text{COST}(\text{Antigen}) \quad (5.2-1)$$

(3) 产生初始抗体群

抗体群记为 *AntigenSet*。其中的每个元素 *AntigenSet*[*i*] ($i \in 0 \sim \text{PopScale} - 1$ ，

$PopScale$ 是抗体群规模) 即 $Antigen$ 采用随机产生的方式: 将一个随机整数 $RAND(RAND \in 0 \sim |P| \times k\% - 1)$ 赋给 $Antigen[i]$, ($i \in 0 \sim |D| - 1$)。

(4) 克隆

对抗体群 $AntigenSet$ 中的每一个抗体按如下规模克隆到新的抗体群, 记为 $CloneAntis$ 中:

$$Q[i] = \text{Int} \left[N_c * \frac{\Phi(AntigenSet[i])}{\sum_{j=0}^{PopScale-1} \Phi(AntigenSet[j])} \right] \quad (5.2-2)$$

N_c 为一大于 $PopScale$ 的整数, $\text{Int}[\bullet]$ 表示上取整数。

(5) 变异

变异操作只对新抗体群 $CloneAntis$ 进行而不对 $AntigenSet$ 进行, 按选定的变异概率对每个抗体 $Antigen$ 的每个码位赋一随机值, 与进化算法相比, 克隆策略的变异概率要大一些。

(6) 克隆选择

选择规则如下:

$$AntigenSet[i] = \begin{cases} CloneAntis[j] & \Phi(AntigenSet[i]) < \Phi(CloneAntis[j]) \\ AntigenSet[i] & \text{其它} \end{cases} \quad (5.2-3)$$

其中, $i \in 0 \sim PopScale - 1$, $j \in \sum_{r=0}^{i-1} Q[r] \sim \sum_{r=0}^i Q[r]$ 。

5.2.2 算法复杂度分析

设网络节点总数为 n , 则搜索备选路径的时间复杂度为 $O(n)$; 对备选路径按代价进行快速排序的时间复杂度为 $O(\log(|P|))$; 求解最优多播树的时间复杂度为 $O(S \times N_c \times |D|)$, S 为算法的迭代次数, 其大小和 N_c 的值都可以被控制为与 n 线性相关, 因此时间复杂度近似为 $O(n^2 \times |D|)$ 。而 BSMA 算法的时间复杂度近似为 $O(|n|^3 \log |n|)$, 可见本算法具有更小的时间复杂度。

5.3 仿真实验

实验比较了本文算法、BSMA 以及 GA^[49]得到的时延受限的多播树的费用。在网络节点数为某一值的情况下，随机产生多个网络拓扑，并分别在上边运行各种算法 50 次，仿真结果取统计平均值。随机网络生成利用了文献[50]中的方法。

实验中，本文算法令 k 的初值等于 6，并取变异概率为 0.5；令 GA 的变异概率为 0.05，交叉概率等于 0.6。两种算法的种群规模都取为网络节点数的 3 倍，克隆策略中的常数 N_c 取为种群规模的 1.5 倍。两种算法均以适应度连续迭代 50 次不变作为算法结束的条件。同时，为了测试 BSMA 得到的多播树的最小费用，仿真时并不限制其替换超边的数量。

从图 5.2~5.4 可以看出，CS 所得的多播树的费用明显小于其他两种算法。图 5.6 是 GA 和 CS 作用于图 5.5 所示网络时的迭代情况的比较，算法开始时，GA 所得的多播树费用较小，迭代到 40 代以后，GA 与 CS 的比值就逐渐变大，说明 GA 陷入了早熟，而 CS 则找到了代价更小的多播树。

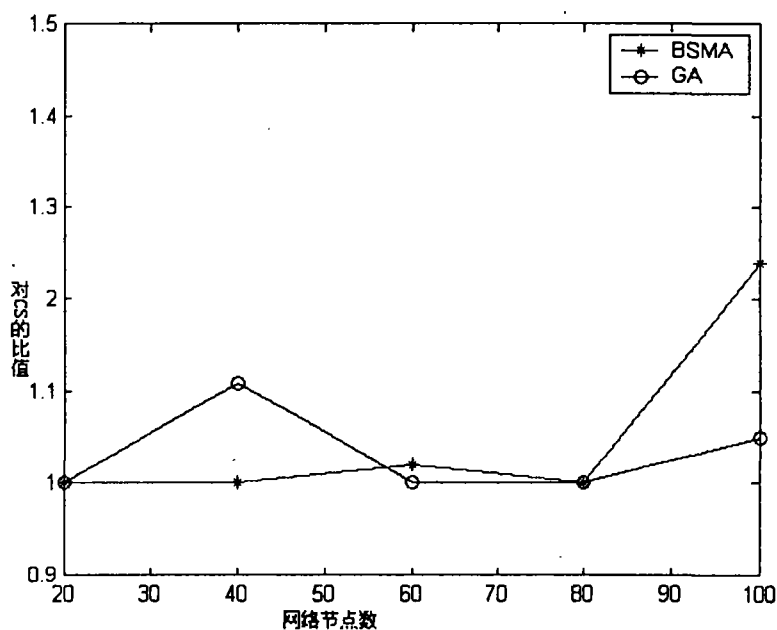


图 5.2 目标节点占网络节点总数 5% 时，GA、BSMA 所得多播树的费用和 CS 的比值

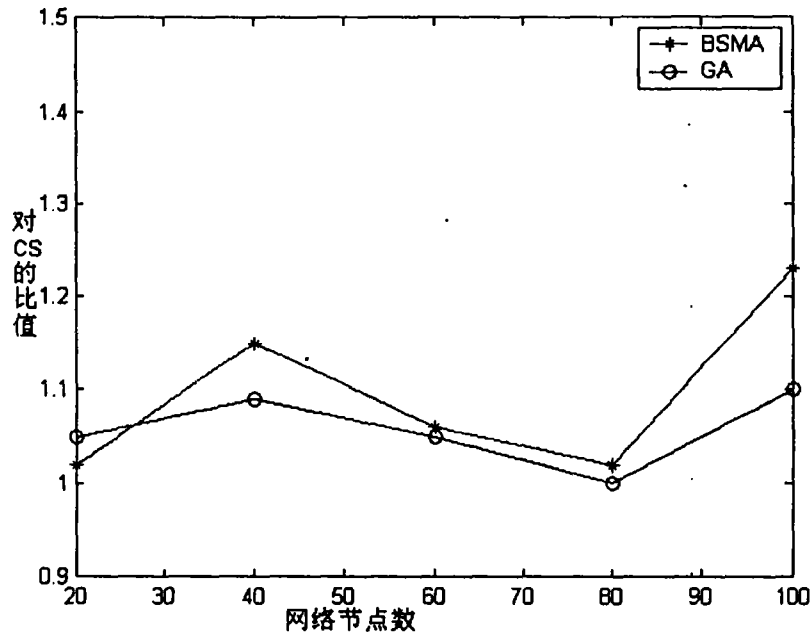


图 5.3 目标节点占网络节点总数 15% 时, GA、BSMA 所得多播树的费用和 CS 的比值

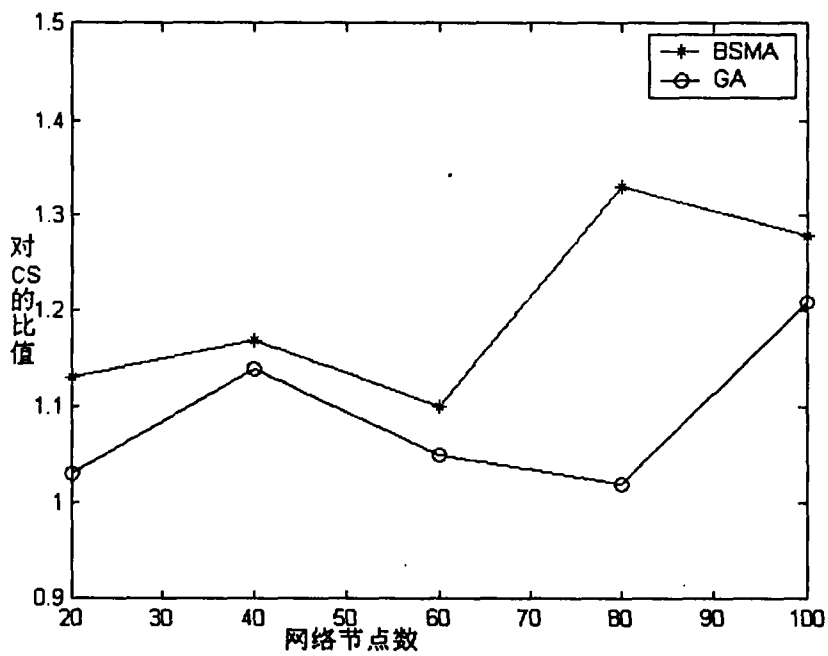


图 5.4 目标节点占网络节点总数 30% 时, GA、BSMA 所得多播树的费用和 CS 的比值

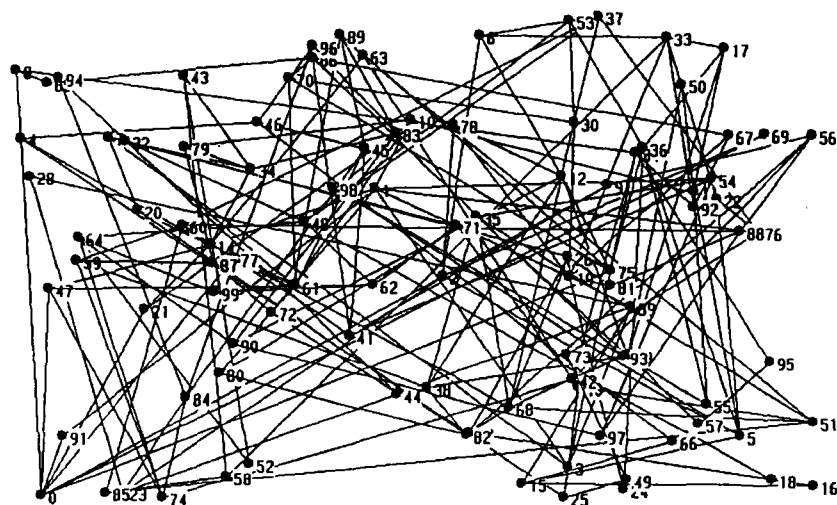


图 5.5 100 节点的随机网络

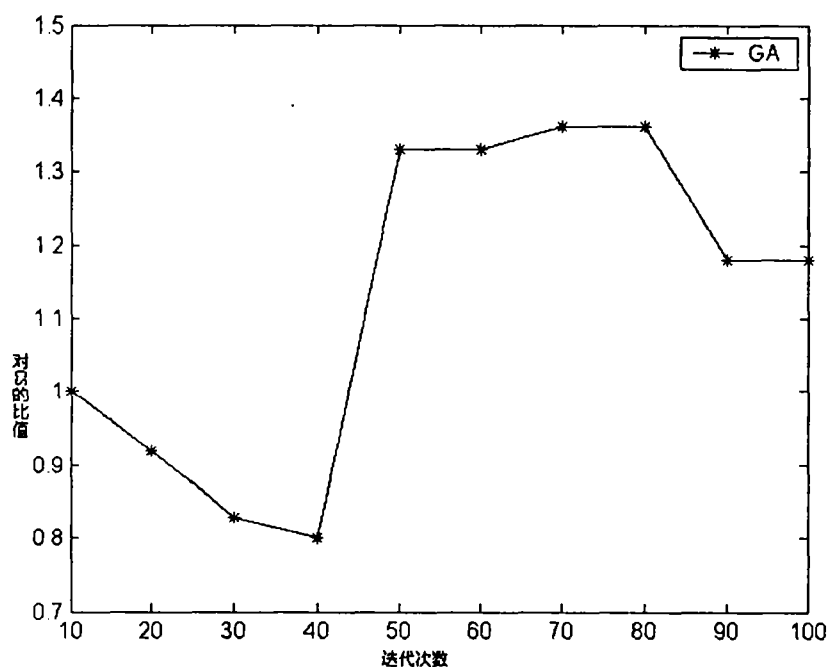


图 5.6 目标节点占网络节点总数 30% 时, GA 所得多播树的费用和 CS 的比值

5.4 结论

本文利用克隆策略解决多播路由问题, 克服了遗传算法中的“早熟”现象, 取得了比确定性算法和遗传算法更好的结果。并且通过引入一个可调参数缩小了备选路径集, 加快了算法的收敛速度。但当备选路径集庞大时, k 值的选取就会变的很困难, 目前本文还未能给出相应的理论推导。

结束语

计算机网络飞速发展，网络功能日益增大。网络的作用从简单的信息传送发展到远程教学。视频会议、数据分发和网络游戏等等，用户的数据要从一个终端发送到另一个终端。首先要确定传输路径。不同的通信方式确定路由的方式也不同。多播是一主机向多个主机发送信息（但不是所有主机）的通信方式，涉及多播技术的应用很多，如多媒体会议、远程教育、数据分发等。多播是目前研究得最多，也是应用最广的网络连接方式。多播问题的关键在于多播路径的确定，即最优多播树的建立。本文主要研究时延受限的多播路由算法，如何建立满足约束条件，费用最低的多播树。

由于求解费用最小多播树在数学上归结为 Steiner 树问题，为此，第二章介绍了有关 Steiner 树问题的理论。其中包括求解 Steiner 树问题的启发式算法、算法的复杂度和性能分析。为了分析算法的平均性能，要进行仿真实验。我们介绍了仿真多播算法普遍采用的随机网络模型，它由北卡罗来那州立大学的 Salama 和 Reeves 在 Waxman 网络的基础上提出的，与 Waxman 网络相比，该网络模型能更好的反应用户对节点的平均度的要求，并且绝对不会有孤立的节点产生，文中算法都用此模型进行实验。

接着，第三章提出了一种基于克隆策略的多播路由算法。克隆策略的实质是在进化中，在最优解的附近产生一个变异解的群体，从而扩大了搜索范围，并增加了抗体的多样性，有助于防止进化早熟搜索陷入局部最优。

然后，给出了免疫克隆多播路由算法。在基于免疫克隆策略的基础上有效地解决了多播路由中的全局优化问题。与传统的遗传算法相比，本算法通过引入与具体问题相关的先验知识来指导搜索过程，加速算法的收敛速度。并且本算法操作简单，易于实现。经过仿真证明，对于一定规模的多播问题，本算法具有较快的收敛速度和较高的准确率。

最后，在第五章中介绍了参数可调的克隆多播路由算法，该算法通过引入一个可调因子缩小了搜索空间，加速了算法的收敛速度。算法实现简单、控制灵活。仿真表明，该算法的性能优于 BSMA 算法和传统的遗传算法。

本文主要研究了时延受限的多播路由算法，提出了几种有效的多播路由算法。但还有很多问题有待今后研究，比如说，如何在本文所提出的几种算法使用二进

制编码求解问题；如何更有效的结合启发算法和免疫算法等。

致谢

本文是在我的导师刘芳教授的亲切关怀和悉心指导下完成的，导师渊博的专业知识，深厚的学术功底和独到的见解，严谨的治学态度、孜孜不倦的钻研和勤奋进取的精神，平易近人的品德，在言传身教中令我受益匪浅，终生难忘。使我不仅学到了许多知识，更重要的是学到了一种治学精神：精益求精、矢志不渝和刻苦钻研的精神，以及严肃认真的做人、做事的态度，这是我一生的财富。两年来我的每一步成长都蕴含随着刘芳教授的心血，在此表示我最诚挚的谢意。

感谢我的同学孙登云、贾凡、杨承勇、谢银祥、郭敬宇、田鹏举、张磊、罗静、李超洋等。在我攻读硕士学位期间，他们给予了我很多帮助，与他们的讨论开阔了我的思路，与他们的相处丰富了我的人生。

感谢同门师兄、师姐，感谢他们在学习上为我提供的帮助和便利。

感谢所有关心、帮助、爱护过我的人们。

谨以此文献给我含辛茹苦养育我，为我的成长倾注了无数心血的父母。

参考文献

- [1] Andrew S.Tnenbaum. Computer Networks.Prentice Hall International,Inc.1998.
- [2] Kadaba Baharath-Kunar.Routing to multiple destinations in computer networks. IEEE Trans on Commun. 1983,COM-301:343-351
- [3] F. Hwang and D. Richards, Steiner Tree Problems. Networks, vol. 22, no. 1, pp. 55–89, January 1992.
- [4] S. Ramanathan, An AlgorithmforMulticast Tree Generation in Networks with Asymmetric Links. in Proceedings of IEEE INFOCOM '96, pp. 337–344, 1996.
- [5] L. Kou, G. Markowsky, and L. Berman. A Fast Algorithm for Steiner Trees. Acta Informatica, vol. 15, no. 2,pp. 141–145, 1981.
- [6] V. Rayward-Smith, The Computation of Nearly Minimal Steiner Trees in Graphs. International Journal of Mathematical Education in Science and Technology, vol. 14, no. 1, pp. 15–23, January/February 1983.
- [7] H. Takahashi and A. Matsuyama, An Approximate Solution for the Steiner Problem in Graphs. Mathematica Japonica, vol. 24, no. 6, pp. 573–577, February 1980.
- [8] Beau Williamson. Developing IP multicast networks. Cisco Press,1998.
- [9] Dijkstra E.W. A note on two problems in connection with graphs. Numberische Mathamtik 1. 1959:195-207.
- [10]Ballardie T,Francis P,Crowcroft J. Core based trees(CBT).In: An Architectures for Scalable Inter-Domain Multicast Routing .SIGCOMM'93, San Francisco, Sept,13-17,1993:85-95
- [11]A Ballardie. Core based trees multicast routing architecture. RFC 2201. Sepetember 1997.
- [12]D. Thaler and C. Ravishankar. Distributed center location algorithms. IEEE Journal on Selected Areas in Communications, 15(3):291--303, April 1997
- [13]S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L.Wei, An architecture for wide-area multicast routing. in Proc. ACM SIGCOMM, Aug. 1994, pp. 126–135.
- [14]T. Ballardie, P. Francis, and J. Crowcroft, An architecture for scalable inter-domain multicast routing. in Proc. ACM SIGCOMM, Sept. 1993, pp. 85–95.
- [15]Hwa-Chun Lin; Zhe-Hong Lin. Selection of candidate cores for core-based multicast routing architectures. Communications, 2002. ICC 2002. IEEE

- International Conference on , Volume: 4 , 28 April-2 May 2002 pp: 2662 -2666
vol.4
- [16]Moy J. Multicast routing extension for OSPF. Commun.ACM.Aug 1994,37:61-66.
- [17]Deering S,Partridge C,Waitzman D. Distance vector multicast routing protocol. RFC 1075,Nov 1998.
- [18]Thyagarajan A, Deering S. Hierarchical distance-vector multicast routing for the Mbone. In:ACM SIGCOMM'95,Cambridge,September 1995, pp:60-66.
- [19]Deering.S,etal. Protocol independent multicast(PIM):dense mode protocol specification.Internet Draft,Mar 1994.
- [20]Deering.S,etal. Protocol independent multicast(PIM):sparse mode protocol specification.Internet Draft,Mar 1994.
- [21]R. Bellman, Dynamic Programming. Princeton University Press, 1957.
- [22]D. Bertsekas and R. Gallager, Data Networks. Prentice-Hall, 2nd ed., 1992.
- [23]B. Awerbuch, A. Bar-Noy, and M. Gopal, Approximate Distributed Bellman-Ford Algorithms. in Proceedings of IEEE INFOCOM '91, pp. 1206-1213, 1991.
- [24]D. Bertsekas, Linear Network Optimization: Algorithms and Codes. MIT Press,1991.
- [25]Y. Dalal and R. Metcalfe, Reverse Path Forwarding of Broadcast Packets. Communications of the ACM, vol. 21, no. 12, pp. 1040-1048, December 1978.
- [26]S. Deering, Multicast Routing in a Datagram Internetwork. PhD thesis, Stanford University, December 1991.
- [27]S. Deering and D. Cheriton, Multicast Routing in Datagram Internetworks and Extended LANs. ACM Transactions on Computer Systems, vol. 8, no. 2, pp. 85-110, May 1990.
- [28]R. Karp, Reducibility among Combinatorial Problems. in Complexity of Computer Computations (R. Miller and J. Thatcher, eds.), pp. 85-103, Plenum Press, 1972.
- [29]D. Wall, Selective Broadcast in Packet-Switched Networks. in Proceedings of the Sixth Berkeley Workshop on Distributed Data Management and Computer Networks, pp. 239-258, February 1982.
- [30]D. Wall, Mechanisms for Broadcast and Selective Broadcast. PhD thesis, Stanford University, June 1980.
- [31]M. Doar and I. Leslie, How Bad is Naive Multicast Routing. in Proceedings of IEEE INFOCOM '93, pp. 82-89, 1993.
- [32]M. Doar, Multicast in the Asynchronous Transfer Mode Environment. PhD thesis, University of Cambridge, January 1993.

- [33] V. Rayward-Smith and A. Clare, On Finding Steiner Vertices. *Networks*, vol. 16, pp. 283-294, 1986.
- [34] X. Jiang, Routing Broadband Multicast Streams. *Computer Communications*, vol. 15, no. 1, pp. 45-51, January, February 1992.
- [35] X. Jiang, Distributed Path Finding Algorithm for Stream Multicast. *Computer Communications*, vol. 16, no. 12, pp. 767-775, December 1993.
- [36] C.H. Chow, On Multicast Path Finding Algorithms. in *Proceedings of IEEE INFOCOM '91*, pp. 1274-1283, 1991.
- [37] Y.W. Leung and T.S. Yum, Efficient Algorithms for Multiple Destinations Routing," in *Proceedings of the IEEE International Conference on Communications (ICC '91)*, pp. 1311-1317, 1991.
- [38] F. Bauer and A. Varma, Distributed Algorithms for Multicast Path Setup in Data Networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 181-191, April 1996.
- [39] 迟彩霞, 吴士泉 等. 多播路由算法及其在 QoS 路由中的应用. *计算机学报*, 2001, 23 (7) : 718-719.
- [40] 王兴伟, 王志军等. 基于服务质量的多媒体通信初始路由建立算法的研究. *计算机学报*, 2001, 24 (8) : 830-837.
- [41] 王应征, 石冰心. 基于启发式遗传算法的 QoS 组播路由问题的求解. *计算机学报*, 2001, 24 (1) : 55-56
- [42] Wang Z, Crowcroft J. Quality of service for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 1996, 14(7):1228-1234.
- [43] Zadeh L.A. Roles of soft computing and fuzzy logic in the conception, design and deployment of information/intelligent systems. *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Application*, 1998:1-9.
- [44] D E Goldberg. *Genetic algorithm in search, optimization and machine learning*. Reading MA: Addison Wesley. 1996.
- [45] Z. Micalwicz. *Genetic algorithm+data structure=evolution program*. 3rd Ed. New York: Springer-Verlag. 1996.
- [46] 刘莹. 计算机网络中的多播路由算法. [博士论文] 西安电子科技大学, 2002.12
- [47] Quan Sun. A Genetic Algorithm for Delay-constrained Minimum-Cost Multicast in.. Institute of Operating Systems and Computer Networks Technical University of Braunschweig, Germany.
- [48] Quan Sun and Horst Langendorfer. A new distributed routing algorithm with end-to-end delay guarantee. *IWQoS'97*, May 1997.

- [49]王新红, 王光兴. 基于遗传算法的时延受限代价最小组播路由选择方法[J] 通信学报 2002, 23(3):112-117
- [50]B.M.Waxman. Routing of multiple connections . IEE Trans on commun. 1988 ,6(9)1671-1622.
- [51]丁永生,任立红.人工免疫系统:理论与应用[J].模式识别与人工智能, 2000, 13(1): 52~59
- [52]Jon Timmis, Mark Neal, John Hunt: An artificial immune system for data analysis[J], BioSystems, 2000 (55) :143~150
- [53]王磊, 潘近, 焦李成. 免疫算法[J]. 电子学报, 2000, 28(7): 74~78
- [54]Deaton R., Garzon M., Rose J.A. etal. A DNA based artificial immune system for self-nonsel self discrimination[C]. 1997 IEEE International Conference on Computational Cybernetics and Simulation, 1997, 1: 862 ~866
- [55]Kuznetsov V.A., Knott G.D., Ivshina A.V. Artificial immune system based on syndromes-response approach: recognition of the patterns of immune response and prognosis of therapy outcome[C]. 1998 IEEE International Conference on Systems, Man, and Cybernetics, 1998, 4: 3804 ~3809
- [56]Sasaki M., Kawafuku M., Takahashi K.. An immune feedback mechanism based adaptive learning of neural network controller[C]. ICONIP '99. 6th International Conference on Neural Information Processing, 1999, 2: 502~507
- [57]H.Meshref, H.VanLandingham. Artificial immune systems: application to autonomous agents[C]. 2000 IEEE International Conference on Systems, Man, and Cybernetics. 2000, 1: 61~66
- [58]D Dasgupta, S.Forrest. Artificial immune systems in industrial applications[C]. IPMM '99. Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials. 1999, 1: 257~267
- [59]J. Timmis, M. Neal, J. Hunt. Data analysis using artificial immune systems, cluster analysis and Kohonen networks: some comparisons[C]. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics. 1999, 3: 922~927
- [60]K.K. Kumar, J. Neidhoefer. Immunized adaptive critics for level 2 intelligent control[C]. Computational Cybernetics and Simulation., 1997 IEEE International Conference on Systems, Man, and Cybernetics. 1997, 1: 856~861
- [61]Jin-Hyung Jun, Dong-Wook Lee, Kwee-Bo Sim. Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system[C]. IEEE SMC '99 Conference Proceedings. 1999 IEEE

- International Conference on Systems, Man, and Cybernetics. 1999, 6: 614~619
- [62]D. Dasgupta. Artificial neural networks and artificial immune systems: similarities and differences[C]. Systems, Man, and Cybernetics, 1997 IEEE International Conference on Computational Cybernetics and Simulation. 1997, 1: 873~8781
- [63]K. D.Cooper, M.W.Hall, K.Kennedy. Proceedings of the 1992 International Conference on Computer Languages 1991:96~105.
- [64]M.Balazinska, E.Merlo, M.Dagenais, etl.. Advanced clone-analysis to support object-oriented system refactoring [C]. Proceedings. Seventh Working Conference on Reverse Engineering,2000:98~107.
- [65]N.Esmaili, C.Sammut, G.M.Shirazi. Behavioural cloning I control of a dynamic system [C]. IEEE International Conference on Systems, Man and Cybernetics Intelligent Systems for the 21st Century. 1995,3:2904~2909.
- [66]M.Hybinette, R.Fujimoto. Cloning: A Novel Method for Interactive Parallel Simulation[C].Proceedings of the 1997 Winter Simulation Conference, 1997:444~451.
- [67]L. N.D.e Castro, F.J.Von Zuben. The Clonal Selection Algorithm with Engineering Applications[C],Proc. Of GECCO'00,Workshop on Artificial Immune Systems and Their Applicatins,2000:36~37
- [68]Jungwon Kim, P.J. Bentley. Towards an artificial immune system for network intrusion detection: an investigation of clonal selection with a negative selection operator[C]. Proceedings of the 2001 Congress on Evolutionary Computation, 2001, 2:1244~1252.
- [69]Haifeng DU, Licheng JIAO, Sun'an Wang. Clonal Operator and Antibody Clone Algorithms. Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 4~5 November 2002:506~510.
- [70]刘若辰, 杜海峰, 焦李成. 免疫克隆策略。[研究报告] 西安, 西安电子科技大学雷达信号处理国家重点实验室与智能信息处理研究所, 2002 年。
- [71]Mehrddad Parsa. An interative algorithm for delay-constrained minimum-cost multicasting. IEEE/ACM Trans On Networking. Vol.6, No.4, pp.461-474, August 1998.
- [72]Hussein F Salama. Evaluation of multicast routing algorithm for real-time communication on high-speed networks.IEEE J on Sel. Areas in Comm,Vol.15, No.3, pp.332-345, Apr.1997.
- [73]V P Kompella. Multicast routing for multimedia communication.IEEE/ACM Trans on Networking. 1993,1:286-292.

- [74]Chor Ping. Loop-free multicast routing with end to end delay constraint. Computer Communications. 1999,22:181-192.
- [75]J.D.Bagley. The behavior of adaptive systems which employ genetic and correlation algorithms. Dissertation Abstracts International,1968,28(12)
- [76]R.S.Rosenberg. Simulation of genetic populations with biochemical properties. Dissertation Abstracts International, 1968,28(7)
- [77]J.H.Holland. Genetic algorithms and classifier systems: foundations and their applications.Proceedings of the Second International Conference on Genetic Algorithms,1987:82~89
- [78]J.J.grefenstette,et al. Genetic algorithms for the traveling salesman problems. Proceedings of an International Conference on Genetic Algorithms and Their Applications,1985:136~140
- [79]D.B.Fogel. Applying evolutionary programming to selected traveling salesman problems. Cybernetics and Systems,1993(24):27~36
- [80]G.Miller,etal. Designing neural networks using genetic algorithm. ICGA,1989:360~369
- [81]Booker,L.B.Goldberg and J.H.Holland. Classifier systems and genetic algorithms. Artificial Intelligence,1989(40):235~282
- [82]J.D.Farmer, N.H.Packard and A.S.Perelson. The immune system. Adaptation and Machine learning.Physics 22D,1987
- [83]H.Bersini and F.J.Varela.Hints for adaptive problem solving gleaned from immune networks. Proceedings of the first workshop on parallel problem solving from nature.Springer &Verlag,1990
- [84]陈国良, 王煦法, 庄镇泉, 王东升. 遗传算法及其应用. 北京: 人民邮电出版社, 1996
- [85]G.Rudlph. Convergence analysis of canonical genetic algoritms. IEEE Trans. On Neural Networks.1994,5(1):96~101
- [86]陈仁. 免疫学基础, 北京: 人民卫生出版社, 1982
- [87]张讲社, 徐宗本, 梁怡. 整体退火遗传算法及收敛充要条件. 中国科学(E), 1997, 27(2):154~164
- [88]Stenfan Vob. Steiner's problem in graphs:heuristic methods. Discrete Applied Mathematics. 1992, 40(1): 43-72.
- [89]石坚, 邹玲, 等. 遗传算法在组播路由选择中的应用. 电子学报, 2000, 28(5):88-89
- [90]王磊. 免疫进化计算理论及应用. [博士论文]. 西安电子科技大学, 2001.9.

- [91] R. Widyono, The Design and Evaluation of Routing Algorithms for Real-Time Channels, Technical Report Tr-94-024, Tenet Group, Department of EECS, University of California at Berkeley, 1994
- [92] A. G. Waters, A new heuristic for ATM multicast routing, 2nd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, 1994.
- [93] 杜海峰. 免疫克隆计算与人工免疫网络研究与应用。[博士后研究报告]. 西安电子科技大学, 2003.11.

作者在读期间撰写的论文

- [1] 杨海潮 刘芳. 免疫克隆多播路由算法. 《中国人工智能进展》(2003)
pp. 1232-1237. 北京邮电大学出版社 2003 年
- [2] 刘芳 杨海潮. 一种基于克隆策略的多播路由算法. 《电子与信息学报》 (已
录用)
- [3] 刘芳 杨海潮. 参数可调的克隆多播路由算法. 《软件学报》 (已录用)