第35卷 第18期 计 算 机 工 程 2009年9月
Vol.35 No.18 Computer Engineering September 2009
·博士论文· 文章编号：1000—3428(2009)18—0009—04 文献标识码：A 中图分类号：TP393

自相似结构化覆盖网组播模拟器设计

程 伟[1]，吴产乐[1,2]，叶 刚[1]，程 实[1]，乐 俊[1]
(1. 武汉大学计算机学院，武汉 430079；2. 国家多媒体软件工程技术研究中心，武汉 430072)

摘 要：针对当前流行覆盖网组播模拟器无法准确为组播系统建模及评测性能的问题，提出一种节点群支持整体代换和多树结构的自相似结构化覆盖网组播(S3M)模型的，设计一个结构化覆盖网组播模拟器，在简化S3M模型结构的基础上为其建立合适的仿真模型。与NICE协议的对比表明，该结构化覆盖网组播模型是有效的。

关键词：应用层组播；覆盖网；模拟器；NICE协议；扩展性

Design of Self-similar Structured Overlay Network Multicast Simulator

CHENG Wei[1], WU Chan-le[1,2], YE Gang[1], CHENG Shi[1], YUE Jun[1]
(1. School of Computer, Wuhan University, Wuhan 430079; 2. National Engineering Research Center for Multimedia Software, Wuhan 430072)

【Abstract】Aiming at the problem that various extending overlay simulator can not afford precise modeling and evaluation of the multicast system, this paper proposes Self-Similar Structured Multicast(S3M) whose nodes group supports unitary substitution and multi-tree structure, designs a scheme of structured application layer multicast model simulator. It sets up an appropriate simulation model for S3M based on the sampling scheme of that multicast model. The simulator can substantiate the validity of the S3M protocol in contrast with NICE protocol.

【Key words】application layer multicast; Overlay Network(ON); simulator; NICE protocol; scalability

1 概述

应用层组播的基本模型是在IP组播、覆盖网(Overlay Network, ON)和对等网(Peer to Peer, P2P)等技术基础上形成并发展起来的。应用层组播独立于具体的网络设备，数据复制、转发等功能都由成员主机完成，成员主机之间建立一个叠加在底层网络(Underlay Network, UN)之上实现组播业务逻辑的功能性网络，即覆盖网。覆盖网的结构是影响应用层组播协议性能的重要因素，很多应用层组播协议的评价标准都与其紧密相关。

目前已经提出的应用层组播协议主要有ESM[1], NICE[2], SplitStream[3]等。研究者对这些协议进行仿真分析所采用的仿真工具、仿真场景及评价指标没有统一的标准，这给协议间的性能比较带来了困难[4-6]。研究分析表明，应用层组播协议主要包含组播组成员的管理以及成员之间的数据传输2个方面，组播成员的管理可以通过仿真覆盖网结构进行模拟，成员间的数据组播通过仿真流量进行近似模拟。

2 自相似结构化覆盖网组播的基本概念

在自相似结构化覆盖网组播(Self-similar Structured Multicast, S3M)的相关理论和技术研究中发现，通常认为的“网状模型必定具备组播可靠性”是一种误解，例如NICE采用的中心节点分发的网状拓扑并没有摆脱树状拓扑的局限性。实际上，树状模型是结构化模型中的一类，而网状模型既可以是结构化的，也可以是非结构化的。传统理论常以树状、网状、环状等类似的静态拓扑作为覆盖网的分类方法，因此，覆盖网节点的组织和优化方法是覆盖网拓扑最本质的要素，即“方法”而非“形状”决定了一个覆盖网结构的基本性质。本文定义了一些约束条件从而使分类较为清晰，图1表达了这种分类方法，其中，S3M是结构化覆盖网中正则结构化的一个特例。S3M模型通过自相似结构的代换可获得更好的连接带宽，以实现高质量的应用层组播。这种代换是区别于其他应用层组播模型的关键。图2示意了节点、节点群及其代换操作。

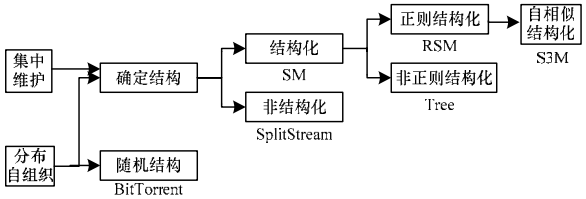

图1 S3M模型与其他应用层组播模型的关系

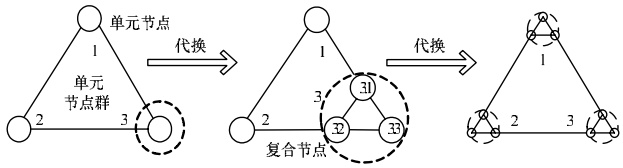

图2 S3M模型的节点代换

3 模拟器设计

3.1 网络实体模型

网络实体既包括现实网络中的主机、链路等，也包括用于实现通信的协议和应用程序。受ACA(Autonomous Component Architecture)网络仿真框架的启发，将网络实体模

基金项目：国家自然科学基金资助项目(60672051)；国家“863”计划基金资助项目(2003AA1032)

作者简介：程 伟(1971－)，男，博士研究生，主研方向：网络通信；吴产乐，教授、博士生导师；叶 刚、程 实、乐 俊，博士研究生

收稿日期：2009-02-20 **E-mail：**wuchl@whu.edu.cn

型主要分为流量、协议、链路、传输、节点 5 大模块，本文研究主要集中在 Overlay 层及 Application 层，它们之间的关系如图 3 所示。

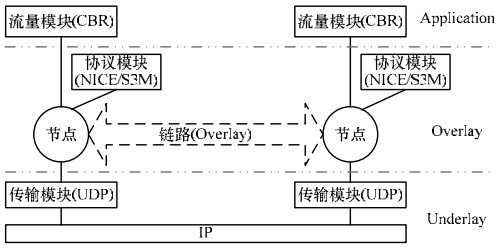

图 3　网络实体模型

(1)流量模块

流量模块相当于实际网络中的应用，其主要功能是：以一定方式产生信息量，交给下层转发；接收来自下层的数据，进行相应处理。数据流的产生方式参照 NS-2 采用的恒定比特流(CBR)流量模型，同时作为可选项，在数据分组发送的时间间隔中可加入随机的抖动时间，主要参数如下：

1)rate：数据发送速率；

2)interval：(可选项)数据块发送的时间间隔；

3)packet_size：产生的数据块大小；

4)random：标志是否需要在发送数据的时间间隔中加入抖动时间，缺省值为 on；

5)maxpkts：发送数据块数量的上限。

(2)节点模块

节点模块表示覆盖网中节点，其内部结构如下：

1)身份标志①属性。节点 ID：每一个节点的唯一标识；组播组标识：节点加入的组播组标识，缺省为 0(没有加入任何组播组)，当节点加入某一组播组时，被设置为该组播组的标识。②方法。身份查询：提供身份查找功能，可返回节点 ID 或地址等信息，也可查询指定身份是否与该节点相符等；身份设置：加入或删除一个身份标志。当身份发生改变时，通知上层协议，以便执行相应操作。

2)邻居表①属性。每个节点都包含一个邻居表信息，该表的表项为：本节点端口号；链路指针；通过该端口可到达的邻居节点指针；邻居节点端口号。②方法。邻居表查询；静态设置邻居表(缺省)；广播方式动态建立邻居表。

3)转发表①属性。发送源节点地址；列表：列表的每一项都包含发送源节点输出端口、端口状态、状态计时器。②方法。路由查找：确定数据的转发路径；路由设置：添加、删除、修改转发表条目。

4)数据报文发送

为上层提供数据的发送服务，当一个数据报文到达时，决定把数据报文转到哪里(一个上层应用或者一个或多个输出端口)。如果数据报文需要转发，将通过调用路由查找得到输出端口列表。

5)组播协议指针

组播协议指针是指向该节点上运行的组播协议模块。

(3)链路模块

现实网络中链路特性参数很多，在模拟中只考虑时延和带宽这 2 个参数。本文只研究主机节点之间的覆盖网链路，其内容包括：链路标号；所连 2 个节点的标号；时延(由拓扑产生时的链路权值确定)；带宽(带宽的设定以一定的分布函数设定)。

端到端时延为链路传输时延、排队时延、主机处理时延之和。链路传输时延与链路带宽及路由密切相关。排队时延是数据包经过各个路由器的缓冲时延总和，不仅取决于当前缓存的数据包数量(为当前网络负载的函数)，还取决于该数据包要经过的输出链路带宽。主机处理时延是两端主机所需处理时间的总和，主要用于数据复制及编解码处理。为简便起见，忽略排队时延，并假设链路每跳传输时延为 D，两端主机处理时延为 $2D$，由此可得主机节点 i 和 j 之间时延 $Delay[i][j]=(|x[i]-x[j]|+|y[i]-y[j]|+2)D$。

同时观察到，链路可用带宽与主机节点距离呈反比关系，假设最大带宽为 B，则节点 i 和 j 之间可用带宽 $Bandwidth[i][j]=B/(|x[i]-x[j]|+|y[i]-y[j]|+n)$，其中，$n$ 为常量。

路径伸展率计算中使用时延参数；数据传输率计算中主要使用链路带宽/节点数据发送率参数。流量模块中 rate(数据发送速率)、interval(数据块发送的时间间隔)、packet_size(产生的数据块的大小)3 个参数用 B/D 表达。

(4)协议模块

这里的协议是指各种需要模拟的应用层组播协议，如 NICE 和 S3M。每个节点都有一个组播协议模块，各种不同的协议根据各自不同的算法设置节点中的转发表，对组播的转发树进行管理。

为简便起见，讨论组播组成员数为 9 的 S3M 组播，随机生成 1 个源节点(中心点)和 9 个接收节点，见图 4。

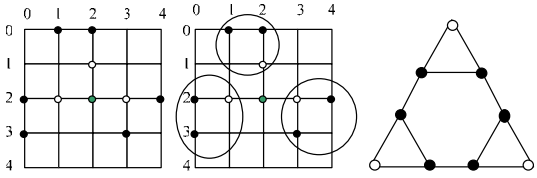

图 4　S3M 仿真实例

1)NICE 实现①n 个节点执行聚合算法，每个节点群节点个数为 K~$3K$-1；②求聚合得到的节点群质心(即各节点的坐标中心点)；③选择离质心最近的节点作为该节点群的 leader；④重复执行①~③，直至选出 $\log_K n$ 层的所有 leader，生成组播转发树。

2)S3M 实现①根据随机生成的 9 个节点与源节点距离得到下载带宽，并从大到小排序，前 k(此处 k=3)个为 S3M 的边界节点(空心点，指直接与源节点连接的接收节点)；②以边界节点为中心进行聚合，节点群中节点个数为 3；③计算各节点群内部节点(实心点)之间带宽 max 值，并对内部节点连接进行调整(即节点代换)，从而生成稳定的转发数据的多树结构，且为正则多树结构。

选择 S3M 边界节点算法如下：

输入　节点 x 坐标数组 $x[i]$，节点 y 坐标数组 $y[i]$，i=0, 1, 2, …, n，其中，节点 0 为源节点

输出　k 个边界节点

Begin

初始化距离数组 d[i][0]为全 0

计算各节点 i 与源节点 0 的距离，计入距离数组 d[i][0]=|x[i]-x[0]|+|y[i]-y[0]|+2d[i][0]值从小到大排序，输出前 k 个 i 的值

End

S3M 节点聚合算法如下：

输入　节点 j

输出　(k-1)个内部节点

Begin

初始化距离数组 d[i][j]为全 0

计算各节点[i]与节点 j 的距离，计入距离数组 d[i][j]=|x[i]-x[j]|+|y[i]-y[j]|+2d[i][j]值从小到大排序，输出前(k-1)个 i 的值

End

(5)传输模块

组播假定一对多的通信方式，不使用 TCP 的端到端机制。UDP 的主要作用是将数据展示给应用层(Overlay 及 Application 层)，并从应用层接收数据，如有需要，进行分段，组成 UDP 报文，再通过网络层单播送出。为便于跟踪分析，模拟系统 UDP 数据报文加上序列号和时间标志。

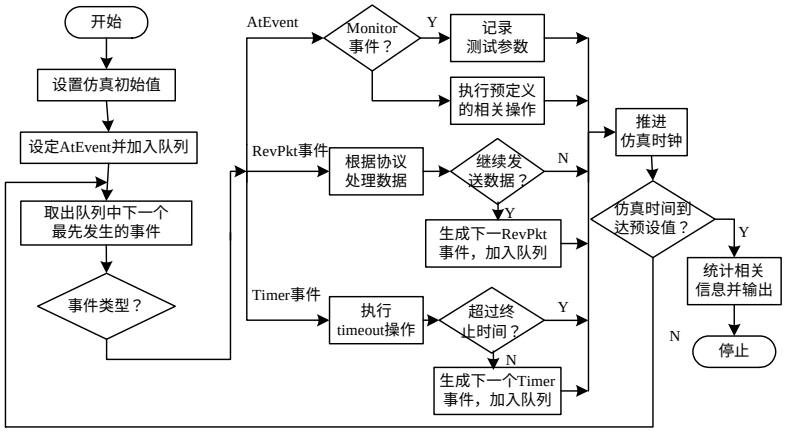

图 5　仿真流程

3.2　调度模型

(1)模拟器

模拟器由离散事件模拟现实网络中数据分组(packet)的流动，包括其在网络设备间的转发以及在网络设备内部的处理过程。

系统由调度器(scheduler)管理一个事件(event)队列，每一个事件单元包含：事件 id(全局唯一)，发生时间，事件处理句柄。其中，事件处理句柄是关键参数，模拟器根据其执行相关操作。调度器根据事件发生时间的先后顺序将事件加入到事件队列中处理，在任何一个给定时刻只有队首事件可被执行。

(2)仿真事件

1)AtEvent 事件

这种事件是在仿真开始运行前设定的，其句柄包含预先定义的事件类型，不同的事件类型对应不同的操作。事件类型分为仿真的开始与结束、各种流量的开始与结束、节点的加入与离开、跟踪监测等。

2)RevPkt(数据报文或控制报文接收)事件

Packet 是系统仿真环境中 2 个对象之间交换的基本单位。由于链路上时延和带宽的影响，同一个节点同时发送出去的数据分组到达所有下游节点的时间有先后，因此，对这些数据的处理也有先后。此外，仿真过程中随时会有一些控制信息(作为特殊的 packet)产生。因此，当节点收到上游节点发送的数据时，应将 RevPkt 事件加入队列中，当调度器调度到该接收事件时再根据事件句柄处理。

3)Timer(计时器)事件

在仿真系统中，流量的产生时间间隔符合一定的统计分布，当流量模块开始发送数据时，需计算下一发送的时间，加入到调度队列中，以便时间到达时可以执行相关操作。与时间相关的模块同时需要相应的计时器，这些事件的句柄称为 Timer 句柄，此句柄本身需提供加入队列、离开队列的接口，它们实际上调用了调度器的加入与离开操作。

4　仿真流程

模拟器的运行流程如图 5 所示。当仿真开始运行时，首先根据预先设定的信息进行初始化设置(如根据信息源类型产生信息源发送对象、根据协议类型为拓扑图中的每个节点产生一个协议对象)，同时产生相关的 AtEvent 事件(如仿真开始/结束事件、监测事件、节点加入/离开事件)加入事件队列。接下来，调度器开始正式工作，它搜寻事件队列中下一个最先发生的事件，取出该事件，判断其事件类型。

如果事件属于 AtEvent 事件，则看其是否为监测(Monitor)事件，是，则系统在该监测时间点记录测试参数(呼叫成功的节点数、源组对的数目、平均路由表尺度、平均时延等)；如果是其他 AtEvent 事件，则进行相应操作。

如果事件属于 RevPk 事件，则依据具体的组播协议处理该数据。组播协议模块将查找节点的组播转发表，根据转发表向组播树的下游节点转发数据。如果数据为控制报文，组播协议将根据操作规程对组播转发表进行重新设置，并把需要继续发送的控制报文向组播树的上游节点发送。发送出去的数据被上游或下游节点接收，又产生接收节点的 RevPk 事件，加入事件队列。

如果事件属于 Timer 事件，如数据源发送事件，则首先产生一定数量的数据，交给下层的 UDP 处理，并判断该时间点是否超过计时器的终止时间，如果没有超过，则继续根据源类型产生下一时刻的源发送事件，加入事件队列。

调度器不断从事件队列中取出最先发生事件并执行事件句柄定义的操作，直到仿真时间到达预定的仿真结束时间时，模拟器系统将进行最后的统计计算，把输出结果显示在界面上。

5　实验分析

实验中随机撒点(源节点和接收节点)，用 NICE 和 S3M 节点组织方式生成数据转发树，并根据应用层组播性能的平均路径伸展率和最大链路压力 2 项主要指标进行比较分析。

(1)路径伸展率

路径伸展率是指在覆盖网上从源主机到某个接收主机之间的组播路径长度与这两者之间直接单播路径长度的比值，也称为相对时延损耗(Relative Delay Penalty, RDP)。平均路径伸展率计算公式为

$$P_{avg}=\frac{\sum_{r\in R}P_r}{|R|}$$

其中，R 为组播节点集合；P_r 是某个接收节点 r 的路径伸展率；$|R|$ 是组播组中的成员数目。

可得出

$$P_{avg}[S3M]=\frac{k+\sum_{i\in C,j_n\neq i}\dfrac{d_{0i}+d_{ij_1}+d_{ij_2}+\cdots+d_{j_{n-1}j_n}}{d_{0j_n}}}{k\times|R|}$$

其中，k 为 S3M 的度数(即边界节点个数)；$j_1,j_2,\cdots,j_{n-1}$ 为边界节点 i 到任意节点 j_n 路径上的转发节点序列($i\neq j_n$)；d_{0i} 为

边界节点 i 到源节点 0 的距离；d_{0j_n} 为接收节点 j_n 到源节点 0 的距离。

$$P_{avg}[NICE]=\frac{1+\sum_{j_n\in Cluster_i}\frac{d_{0l_m}+d_{1l_m}+\cdots+d_{l_1j_n}}{d_{0j_n}}}{|R|}$$

其中，j_n 为 Layer0 层 $cluster_i$ 中节点；其上层 leader 节点依次为 $\{l_1,\cdots,l_{m-1},l_m\}$。

可得到 $P_{avg}[NICE]>P_{avg}[S3M]$。

图 6 为 30 次随机撒点实验得出的平均路径伸展率，对于该性能指标，S3M 比 NICE 更优。

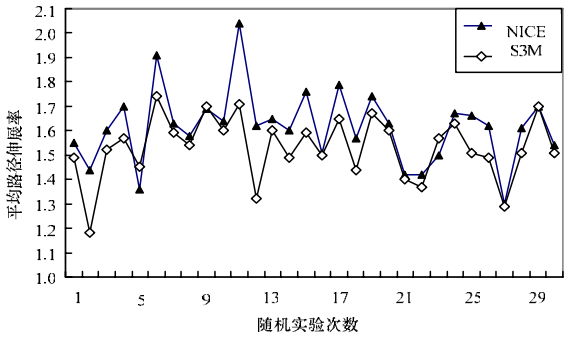

图 6　平均路径伸展率比较

(2)链路压力

链路压力指在某个组播树的链路上发送或接收同一份组播数据的次数。最大链路压力为 $S_{\mathrm{MAX}}=\underset{l\in L}{\mathrm{MAX}}\{S_l\}$，其中，$L$ 为链路集合；S_l 是某条链路 l 的压力。

简化拓扑下，根据 NICE 单树结构和 S3M 自相似结构可知，$S_{\mathrm{MAX}}[NICE]=n\times R_{src}$，$S_{\mathrm{MAX}}[S3M]=k\times\frac{R_{src}}{k}=R_{src}$，其中，$n$ 为 NICE 组播节点个数；k 为 S3M 组播度数；R_{src} 为源节点数据发送率；显然 $S_{\mathrm{MAX}}[NICE]>S_{\mathrm{MAX}}[S3M]$。

6　结束语

本文基于 ACA 标准网络仿真框架设计实现了一个应用层组播仿真系统，其特点是只需编写各种应用层组播协议模块，通过设置不同的网络环境可实现应用层组播的仿真。分析了应用层组播的主要性能指标及计算公式，通过简化实例对 S3M 和 NICE 进行了仿真分析，结果显示 S3M 平均路径伸展率和最大链路压力均小于 NICE，具有较优的组播 性能。

参考文献

[1] Chu Yanghua, Rao S, Sanjay G, et al. A Case for End System Multicast[C]//Proc. of ACM SIGMETRICS'00. Santa Clara, CA, USA: ACM Press, 2000-06.

[2] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable Application Layer Multicast[C]//Proc. of ACM SIGCOMM'02. Pittsburgh, PA, USA: ACM Press, 2002-08.

[3] Castro M, Druschel P, Kermarrec A, et al. SplitStream: High-bandwidth Multicast in Cooperative Environments[C]//Proc. of the 19th ACM SOSP. Bolton Landing, NY, USA: ACM Press, 2003-10.

[4] Naicken S, Basu A, Livingston B, et al. A Survey of Peer-to-Peer Network Simulators[C]//Proc. of the 7th Annual Postgraduate Symposium. Liverpool, UK: [s. n.], 2006.

[5] Shudo K. Overlay Weaver[EB/OL]. [2007-09-01]. http://overlay weaver.sourceforge.net/.

[6] AndrasV. OMNeT++: Discrete Event Simulation System[EB/OL]. [2007-09-01]. http://www.omnetpp.org.

编辑　张正兴

(上接第 8 页)

表 2　EGEA-M 与 5 种算法对 30 维函数 f_1-f_6 优化结果的对比

Function		f_1	f_2	f_3	f_4	f_5	f_6
EGEA	MNFE	28 786	11114	21409	25 858	20 391	9 009
	MFV	-12 569.236 7	0	0	2.5e-4	6.8e-2	0
	(Std)	(1.5e-1)	(0)	(0)	(8.7e-3)	(3.9e-4)	(0)
OGA/Q	MNFE	302 166	224 710	112 421	134 000	112 559	112 612
	MFV	-12 569.453 7	0	4.44e-16	0	0	0
	(Std)	(6.447e-4)	(0)	(3.9e-16)	(0)	(0)	(0)
MAGA	MNFE	10 862	11 427	9 656	9 777	9 502	9 591
	MFV	-12 569.486 6	0	4.44e-16	0	0	0
	(Std)	(7.121e-12)	(0)	(0)	(0)	(0)	(0)
HTGA	MNFE	163 468	16 267	16 632	20 999	20 844	14 285
	MFV	-12 569.46	0	0	0	0	0
	(Std)	(0)	(0)	(0)	(0)	(0)	(0)
StGA	MNFE	1 500	28 500	10 000	52 500	30 000	17 600
	MFV	-12 569.5	4.42e-13	3.52e-8	2.44e-17	2.45e-15	2.03e-7
	(Std)	(0)	(1.1e-13)	(3.5e-9)	(4.5e-17)	(5.2e-16)	(2.95e-8)

在对函数 f_2 和 f_6 的优化中，EGEA-M 使用了最少的函数评估次数找到了全局最优解；在对函数 f_1 的优化中 StGA 是唯一找到最优解的算法，EGEA-M 优化结果的精度不如上述算法；在对函数 f_3 的优化中 MAGA 花费的平均函数评估次数最少，EGEA 与 HTGA 搜索结果的精度最高，且 EGEA 花费的平均函数评估次数要少于 OGA/Q；在对函数 f_4 和 f_5 的优化中 MAGA 表现出了最优的性能，EGEA 使用的函数评估次数虽然要少于 OGA/Q 和 StGA，但优化结果的精度并不理想。综合来说，在对函数 f_1-f_6 的优化中 MAGA 的整体表现最佳，EGEA 与其他 4 种算法针对不同函数的优化效果各有优势。

4　结束语

族群进化算法所特有的群体结构是择偶策略实施的基础，通过对多维函数的优化实验也证明这种繁殖机制的有效性。受此结果启发，针对族群机制对群体中典型个体的筛选和分类能力，设计基于对族群进化知识挖掘和利用的算子将是下一步的研究工作。

参考文献

[1] Chen Hao, Cui Duwu. The Harmonious Evolution of Ethnic Group Algorithm[C]//Proc. of the 3rd International Conference on Natural Computation. Haikou, China: IEEE Computer Society, 2007: 380-384.

[2] Domingo O B, Cesar H M, Nicolas G P. Improving Crossover Operator for Real-coded Genetic Algorithms Using Virtual Parents [J]. Journal of Heuristics, 2007, 13(3): 265-314.

[3] Garcia-Martinez C, Lozano M, Herrera F, et al. Global and Local Real-coded Genetic Algorithms Based on Parent-centric Crossover Operators[J]. European Journal of Operational Research, 2008, 185(3): 1088-1113.

[4] Zhong Weicai, Liu Jing, Xue Mingzhi, et al. A Multiagent Genetic Algorithm for Global Numerical Optimization[J]. IEEE Trans. on Systems, Man and Cybernetics——Part B: Cybernetics, 2004, 34(2): 1128-1141.

[5] Leung Yiuwing Wang Yuping. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization[J]. IEEE Transactions on Evolutionary Computation, 2001, 5(1): 41-53.

编辑　索书志