

求解车辆路径安排问题的混合遗传算法

戴树贵^{1,2},姜昌华¹,潘荫荣¹,胡幼华¹

DAI Shu-gui^{1,2},JIANG Chang-hua¹,PAN Yin-rong¹,HU You-hua¹

1.华东师范大学 计算机科学技术系,上海 200062

2.滁州学院 数学系,安徽 滁州 239000

1.Dept. of Computer Science and Technology,East China Normal University,Shanghai 200062,China

2.Dept. of Mathematics,Chuzhou University,Chuzhou,Anhui 239000,China

E-mail:daisg2008@163.com

DAI Shu-gui, Jiang CHANG-hua, PAN Yin-rong, et al. Hybrid Genetic Algorithm for solving Vehicle Routing Problem. Computer Engineering and Applications, 2007, 43(21): 225-228.

Abstract: Capacitated Vehicle Routing Problem(CVRP) is discussed, and an efficient hybrid genetic algorithm is designed. Crossover and selection rule are designed to overcome the shortcoming of simple genetic algorithm that is easy to trapping in local optimum. Crossover can be performed only between two individuals satisfying the given condition. Better individuals preserved policy is adopted and a function is designed to decide the number of better individuals. Optimal policy is schemed out according to locations relations between points and some individuals are selected to optimize in every generation according to the given probability. The data experiment show the algorithm is an efficient algorithm for solving CVRP.

Key words: Vehicle Routing Problem; genetic algorithm; crossover rule; optimal policy

摘要:讨论了具有容量限制的车辆路径安排问题,设计了一个高效混合遗传算法。针对简单遗传算法易收敛于局部最优解的缺点,算法设计了交叉规则和选择策略。只有当两个个体的评价函数值满足一定条件时,才能进行交叉操作。采用优良个体保留策略执行选择操作,设计了保留函数。算法依据顶点间的位置关系,设计了优化策略,在每代进化中按概率选择一定数量的个体执行优化操作。数据实验表明,该算法是一个有效的求解车辆路径安排问题的混合遗传算法。

关键词:车辆路径安排问题;遗传算法;交叉规则;优化策略

文章编号:1002-8331(2007)21-0225-04 **文献标识码:**A **中图分类号:**TP181;U11

1 引言

车辆路径安排问题(Vehicle Routing Problem, VRP)是物流科学研究的热点问题之一,利用计算机技术解决VRP通常可以使配送费用节省5%~20%。VRP根据限制条件的不同通常可以分为多种类型。本文研究车辆具有容量限制的VRP(Capacitated VRP, CVRP),它包括了运输路线安排问题和车辆调度问题,被认为是一个NP-hard问题。

CVRP描述如下:

一个配送中心有多辆运输车需要将多件货物运往多个客户,要求满足以下约束:(1)每个客户均有车辆提供服务,且仅有一辆车为之服务;(2)每辆车均由配送中心出发并最后回到配送中心;(3)一辆车服务的所有客户的需求量之和不超过该车的载重量;(4)每辆车的行驶距离不超过预定的该车的最大行驶距离。

应该如何安排车辆和运输线路,从而使总的费用最小。文中的费用为车辆行驶距离。

CVRP的求解方法主要有动态规划法、节约法、扫描法、分

区法以及方案评估法等,但是这些确定性研究方法需要花费大量的运算时间,而且随着问题规模的增加,方法将变得不可行或难以得到满意解。近年来,研究者重点研究了运用启发式方法和元启发式算法(Meta-Heuristics algorithm)来解决CVRP。Brandão研究了使用禁忌搜索算法解决CVRP^[1]。Prins给出了一个解决该问题的进化算法^[2]。Mazzeo等使用蚁群算法求解CVRP^[3]。Nagy等研究了解决该问题的启发式算法^[4]。Torki等尝试了使用神经网络方法求解^[5]。郎茂祥给出解决该问题的模拟退火算法^[6]。文献[6-8]使用遗传算法求解CVRP。作者使用单亲遗传算法求解基于实际交通网络的CVRP^[9]。但在现存的基于简单遗传算法设计的解决CVRP的文献,都没有很好地解决简单遗传算法易于收敛于局部最优解(早熟)的问题。

通过对CVRP和遗传算法进行研究,本文为CVRP建立了一个高效的遗传算法,算法设计了选择和交叉规则以及个体优化策略,有效地解决了简单遗传算法的早熟问题并具有较快的收敛速度。数据实验表明算法是有效的。

基金项目:安徽高校省级自然科学基金(No.2006KJ253B)。

作者简介:戴树贵(1973-),男,博士生,研究方向:软件与算法,高级物流;姜昌华(1973-),男,博士生,研究方向:智能算法、物流系统仿真;潘荫荣(1946-),博导,研究方向:物流系统中的计算机应用技术;胡幼华(1946-),博导,研究方向:计算机仿真、物流及供应链管理。

2 问题描述

$G=(V, E)$, $V=\{v_0, v_1, v_2, \dots, v_n\}$ 为顶点集, 其中 v_0 为配送中心, n 为客户数量, $V'=V\setminus\{v_0\}$ 为客户集合. $E=\{e_{ij}=(v_i, v_j)|v_i, v_j \in V, i \neq j\}$ 为边集.

$d_i(1 \leq i \leq n)$ 为客户 i 的需求量, 共有 m 辆运输车, 每辆车的最大载重量为 $W_i(i=1, 2, \dots, m)$, 每辆车的最大行驶距离为 $L_i(i=1, 2, \dots, m)$. $R_k(1 \leq k \leq m)$ 为第 k 辆运输车的运行路线, r_{ki} 表示客户 r_{ki} 在第 k 条路径中的顺序为 i , 规定 $r_{k0}=0$.

$n_k(k=1, 2, \dots, m)$ 表示由第 k 辆车配送的客户的个数, $n_k=0$ 表示未使用该车.

CVRP 的数学模型如下:

$$\min F = \sum_{k=1}^m \left(\sum_{i=1}^{n_k} e_{r_{k(i-1)}, r_{ki}} + e_{r_{kn}, r_{k0}} \right) \quad (1)$$

$$\text{st. } \sum_{i=1}^{n_k} d_i \leq \sum_{j=1}^m W_j \quad (2)$$

$$\sum_{i=1}^{n_k} e_{r_{k(i-1)}, r_{ki}} + e_{r_{kn}, r_{k0}} \leq L_k \quad (3)$$

$$\sum_{i=1}^{n_k} d_i \leq W_k \quad (4)$$

$$\sum_{i=1}^k n_k = n \quad (5)$$

$$R_k = \{r_{ki} | r_{ki} \in \{1, 2, \dots, n\}, i=1, 2, \dots, n_k\} \quad (6)$$

$$R_{k_i} \cap R_{k_j} = \emptyset \quad (7)$$

其中: 式(1)为目标函数, 要求配送车辆行驶路径最短. 式(2)限制客户总需求量不大于车辆的总运输能力. 式(3)式限制每条配送路径长度不大于负责运输的车辆的最大行驶距离. 式(4)限制每辆车所负责的客户的总需求量不大于该车的最大载重量. 式(5)(6)(7)保证每个客户均被服务且只有一辆车为之服务.

3 求解 CVRP 的高效遗传算法

算法假定式(2)自动成立, 否则该问题涉及车辆重复使用问题, 不属于本文的研究范畴.

3.1 预处理

算法首先将所有客户的平面坐标转化成以物流中心为极点、以物流中心和任一客户的连线为极轴的极坐标, 然后对所有客户按极角由小到大的顺序进行排序, 并将排序结果存放于二维数组 $angle[n+1][2]$ 中, 其中 $angle[i][0]$ 为客户极角, $angle[i][1]$ 为客户序号.

3.2 编码方式

采用路径编码方式^[6], 每个个体由区间 $[1, m+n-1]$ 中互不相同的自然数序列构成, 其中 $1 \sim n$ 表示客户, $n+1 \sim n+m-1$ 表示配送中心. 例如, 若有 2 辆运输车和 5 个客户, 则个体为 213654 表示的两条配送路径为: 配送中心 $\rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow$ 配送中心, 配送中心 $\rightarrow 5 \rightarrow 4 \rightarrow$ 配送中心. 该编码方式可以自动保证式(5)(6)(7)成立.

3.3 初始种群的产生

为了使初始种群具有较好的多样性, 本文采用以下三种方式各产生一部分个体来构成初始种群.

(1)在极坐标系中, 分别以每个客户为起始点, 按极角顺序

遍历一周, 在遍历所得的客户序列中, 按约束条件插入表示配送中心的基因(即 $n+1, \dots, n+m-1$), 构成一个个体. 这样可以产生 n 个个体.

(2)重复 $(K-n)/2$ 次以下操作, 产生 $(K-n)/2$ 个个体: 随机产生 n 个区间 $[1, n]$ 上互不重复的自然数, 然后在其中按约束(3)和(4)插入 $n+1, \dots, n+m-1$, 构成一个个体. 若该个体满足所有约束, 则该个体为有效个体, 否则为无效个体, 重新产生该个体.

(3)重复 $(K-n)/2$ 次以下操作, 产生 $(K-n)/2$ 个个体: 随机选定一个客户作为起始点, 运用贪婪算法, 产生一个个体.

3.4 个体评价函数

由于在初始种群中既有有效个体也有无效个体, 后续的遗传操作中也会产生无效个体, 本文对有效个体, 直接使用目标函数值评价个体, 而对无效个体, 引入惩罚数 p , 即个体 i 的评价函数为:

$$F_i = \begin{cases} F_i & (\text{当个体 } i \text{ 为有效个体时}) \\ F_i + p & (\text{当个体 } i \text{ 为无效个体时}) \end{cases} \quad (8)$$

评价函数用于在遗传操作中评价个体的优劣, 评价函数值越小, 则个体越优.

3.5 交叉操作

在交叉个体的选择上, 简单遗传算法通常直接采用随机选择两个个体进行交叉操作. 这种交叉操作很大程度上导致了早熟现象的发生. 为了避免两个相近个体的交配, 本文提出了一种新的交叉规则, 其基本思想是: 在选择交叉个体时, 要求两个个体之间有一定的距离以保持种群的多样性.

对于两个个体 i 和 j , 由式(8)求得的评价函数值分别为 F_i 和 F_j , 定义 $D_{ij} = \frac{|F_i - F_j|}{F_{\max} - F_{\min}}$ 为个体 i 和 j 的范距, 其中 F_{\max} 为该种群中个体的最大评价函数值, F_{\min} 为最小评价函数值. 易知 $0 \leq D_{ij} \leq 1$. 设置阈值 D , 只有两个个体的范距 $D_{ij} \geq D$ 时, 才能进行交叉操作. 若某个个体与其它所有个体的范距均小于 D , 则该个体不与其它个体进行交叉操作.

为了减少交叉过程中产生无效个体的机率, 对选定的两个个体, 首先删除所有表示物流中心的基因, 然后采用基于路径表达的顺序交叉方式^[6]. 例如: 选定的两个个体分别是 32415768119 和 51418372196, 则交叉后形成的两个个体分别是 45618372119 和 14315768129. 最后, 在两个个体中按约束(3)和(4)插入 $n+1, \dots, n+m-1$, 从而构成两个新个体.

算法依据交叉概率, 随机产生需要进行交叉操作的个体数, 然后对随机选定的每个个体, 执行上述操作.

3.6 选择操作

以上交叉个体的选择方案可以很好地解决简单遗传算法的早熟现象, 但是, 它会使得种群进化速度减慢, 导致算法执行时间增加. 为了解决这个问题, 本算法采用优良个体保存策略, 即在产生下一代种群时, 将上一代种群中一定数量的较优个体直接复制到下一代种群中. 由于在种群进化早期, 种群的多样性较好, 可以复制较多的优良个体到下一代种群中, 而到种群进化后期, 种群逐渐收敛, 复制较多的优良个体会使种群快速趋于单一化, 为此可以复制较少的优良种群到下一代种群中. 本文定义的优良个体数函数如下:

$$Op_i = \begin{cases} P \times K & \text{当 } \ln T_i \leq 1 \text{ 时} \\ \lfloor \frac{P \times K}{\ln T_i} \rfloor & \text{当 } \ln T_i > 1, \text{ 且 } \lfloor \frac{P \times K}{\ln T_i} \rfloor > 1 \text{ 时} \\ 1 & \text{当 } \ln T_i > 1, \text{ 且 } \lfloor \frac{P \times K}{\ln T_i} \rfloor \leq 1 \text{ 时} \end{cases} \quad (9)$$

其中: P 为初始确定的优良个体占种群中总个体数的百分比, K 为种群大小, T_i 为进化代数, $\lfloor X \rfloor$ 为不大于 X 的整数。第0代向第1代种群中复制 PK 个体,随着进化代数的逐渐增加,则 Op_i 逐渐减小,直到每代仅复制一个优良个体到下一代种群。

选择操作首先从上一代种群中复制 Op_i 个优良个体到下一代种群中,其余 $K-Op_i$ 个个体采用锦标赛选择方式产生。即做 $K-Op_i$ 次如下操作:产生两个 $[1, K]$ 间均匀分布的随机整数 s 和 t ($s \neq t$),若 $F_s < F_t$,则选择个体 s ,否则选择个体 t 。复制所选择的个体到下一代种群中。

3.7 变异操作

变异操作用以调换同一个体的两个不同位置的基因。首先根据变异概率产生需要进行变异操作的个体个数,然后对于随机选定的每个个体调换两个位置的基因。例如一个个体为456837219,随机产生的两个位置为3和6,则变异的结果产生的新个体为457836219。

对新个体,若为无效个体,则删除所有表示物流中心的基因,然后按约束(3)和(4)重新插入。

3.8 优化策略

在每一代进化中,依据给定的优化概率,选择一定数量的个体进行优化,以加快算法的收敛速度。本文设计了一个两重优化策略:

策略1 优化的基本思想是:在一个个体的两个相邻基因所对应的客户间,按极角顺序插入位于这两个客户间的客户。具体方法如下:

依据优化概率,随机产生需要优化的个体数量 num ,做 num 次以下操作:①随机产生一个 $[1, K]$ 间均匀分布的随机整数 s ;②删除个体 s 中所有表示物流中心的基因;③产生一个 $[1, N]$ 间均匀分布的随机整数 t_1 ,若 $t_1=N$,则 $t_2 \leftarrow 0$ (0表示物流中心),否则 $t_2 \leftarrow G[s][t_1+1]$, $t_1 \leftarrow G[s][t_1]$;④在数组 $angle$ 中查找 t_1 和 t_2 ,它们对应的数组下标分别为 k_1 和 k_2 ;⑤删除个体 $G[s][k_1 \dots k_2]$ 间的所有基因;⑥若 $k_1 > k_2$,则将 $angle[k_2 \dots k_1]$ 按倒序插入到 $G[s]$ 中基因 t_1 之后;⑦若 $k_1 < k_2$,则将 $angle[k_1 \dots k_2]$ 按顺序插入到 $G[s]$ 中基因 t_1 之后;⑧按约束(3)和(4)重新插入表示物流中心的基因。

该优化策略解决运输车辆迂回运输的问题,使车辆按一定方向行驶,但是通常情况下还存在如图1所示的顶点间的关系,这种情况下,按优化策略1车辆行驶的路径为 $a \rightarrow b \rightarrow c \rightarrow d$,显然路径 $a \rightarrow c \rightarrow b \rightarrow d$ 比该路径更优,因此优化策略1并不能保证两点的运输路径最优。为此,本文另使用优化策略2对个体进行优化。

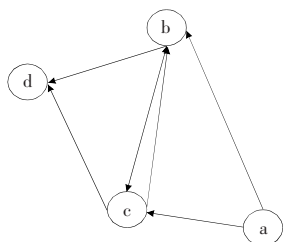


图1 顶点间路径示意图

策略2 使用2-opt优化算法,依据优化概率,随机产生需要优化的个体数量 $num1$,做 $num1$ 次以下操作:①随机产生一个 $[1, K]$ 间均匀分布的随机整数 s ;②产生一个 $[1, N-2]$ 间均匀分布的随机整数 k_1 ;③ $k_2 \leftarrow G[s][k_1+1]$, $k_3 \leftarrow G[s][k_1+2]$;若 $k_1=N-2$,则 $k_4 \leftarrow 0$ (0表示物流中心),否则 $k_4 \leftarrow G[s][k_1+2]$;④ $k_1 \leftarrow G[s][k_1]$;⑤若 $k_1 k_2 + k_2 k_3 + k_3 k_4 > k_1 k_3 + k_1 k_2 + k_2 k_4$,则交换 k_2 和 k_3 的基因位置。

这两重优化策略的实施可以很好地实现配送路径的局部优化。

3.9 算法结束条件

算法在执行一定代数后结束。

4 数据实验

作者使用visual C++6.0设计了程序,分别对VRPNC1和文献[6]中的实例2.1、2.2进行了测试。其中VRPNC1是一个具有50个顶点的通用的CVRP测试数据。结果如下:

(1)对VRPNC1参数如下: $K=160$,进化代数为1600, $P=0.3$,交叉概率为0.5,变异概率为0.02,阈值 $D=0.185$,优化概率为0.2,惩罚函数 p 为1000。算法运行了100次,得到的最好解为524.61,对应的配送方案为:配送中心 $\rightarrow 12 \rightarrow 37 \rightarrow 44 \rightarrow 15 \rightarrow 45 \rightarrow 33 \rightarrow 39 \rightarrow 10 \rightarrow 49 \rightarrow 5 \rightarrow 46 \rightarrow$ 配送中心,配送中心 $\rightarrow 38 \rightarrow 9 \rightarrow 30 \rightarrow 34 \rightarrow 50 \rightarrow 16 \rightarrow 21 \rightarrow 29 \rightarrow 2 \rightarrow 11 \rightarrow$ 配送中心,配送中心 $\rightarrow 32 \rightarrow 1 \rightarrow 22 \rightarrow 20 \rightarrow 35 \rightarrow 36 \rightarrow 3 \rightarrow 28 \rightarrow 31 \rightarrow 26 \rightarrow 8 \rightarrow$ 配送中心,配送中心 $\rightarrow 27 \rightarrow 48 \rightarrow 23 \rightarrow 7 \rightarrow 43 \rightarrow 24 \rightarrow 25 \rightarrow 14 \rightarrow 6 \rightarrow$ 配送中心,配送中心 $\rightarrow 18 \rightarrow 13 \rightarrow 41 \rightarrow 40 \rightarrow 19 \rightarrow 42 \rightarrow 17 \rightarrow 4 \rightarrow 47 \rightarrow$ 配送中心,这也是当前该问题已知的最好解,得到的最差解为537.86,最差解仅比当前已知最好解大2.52%,解的平均值为531.24,比当前已知最好解大1.26%,而运用简单遗传算法所求得的最佳解为831.68。算法平均收敛代数为720,在Intel赛扬2.0GHZ处理器,256M内存,Windows XP操作系统的机器上,算法的执行时间约为4s。所得到解的分布情况如图2所示。其中 $(a, b)X$ 表示算法所求得的大于最好解 a ,而不大于最好解 b 的解的数量占算法执行次数的百分比为 $X\%$ 。

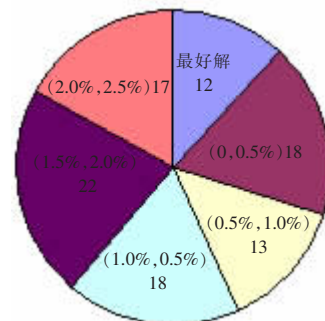


图2 解的分布示意图

一次求得最好解的进化过程如图3所示。其中横坐标为进化代数,纵坐标为每代种群中最优个体的评价函数值,从图3可以看出,算法具有较好的跳出局部最优解的能力。

(2)在未作优化的情况下(即仅采用本文设计的交叉个体选择方案和选择操作方案),对文献[6]实例2.1和2.2所提供的数据进行了测试。其中交叉概率为0.5,阈值 $D=0.15$,种群使用大小为40,进化代数为400,初使优良个体占个体总数的百分比为30%,其它参数和文献[6]相同。在不同随机数序列下,算法执行了100次,实例2.1的结果对比见表1。实例2.2的结果对比见表2。

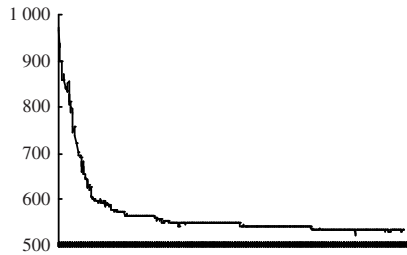


图3 解的进化过程

表1 实例 2.1 结果对比

算法	最优解	最差解	平均解
本文	67.5	69.5	67.7
文献[6]	67.5	71.5	69.7

表2 实例 2.2 结果对比

算法	最优解	最差解	平均解
本文	108.7	146.6	129.7
文献[6]	133.2	147.4	140.1

在 Intel 赛扬 2.0 GHZ 处理器, 256 M 内存, Windows XP 操作系统的机器上, 本算法的执行时间仅为 0.15 s。

从数据实验的结果看, 本文所提供的算法, 很好地解决了简单遗传算法的早熟问题, 并且具有较好的收敛速度, 是一个解决 CVRP 的高效算法。

5 结语

本文为 CVRP 设计了一个高效的遗传算法, 算法针对简单遗传算法因随机选择交叉个体而易于产生早熟现象的问题, 设计了交叉个体选择方案, 从而较好地解决了这一问题。同时为了使算法有一定的收敛速度, 算法采用了优良个体保留方案, 设计了保留函数, 并且针对顶点间的位置关系, 设计了个体优化优化策略, 从而较好地解决了收敛速度问题。数据实验表明, 本文的算法是对解决 CVRP 是高效的。

由于算法的交叉个体选择规则要求两个被选择个体必须有

一定的距离, 因而延缓了算法的收敛速度, 因此为了解决较大规模 VRP, 算法要求使用较大的种群数量和较大的进化代数。

本文的算法虽然是针对 CVRP 设计的, 但是其中的交叉和选择方案, 对于其它 VRP 依然有效。(收稿日期: 2006 年 11 月)

参考文献:

- [1] Brandão J.A tabu search algorithm for the open vehicle routing problem [J].European Journal of Operational Research, 2004, 157: 552-564.
- [2] Prins C.A simple and effective evolutionary algorithm for the vehicle routing problem[J].Computers & Operations Research, 2004, 31: 1985-2002.
- [3] Mazzeo S, Loiseau I.An ant colony algorithm for the capacitated vehicle routing[J].Electronic Notes in Discrete Mathematics, 2004, 18: 181-186.
- [4] Nagy G, Salhi S.Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries [J].European Journal of Operational Research, 2005, 162: 126-141.
- [5] Toriki A, Somhon S, Enkawa T.A competitive neural network algorithm for solving vehicle routing problem[J].Computers & Industrial Engineering, 1997, 33(34): 473-476.
- [6] 郎茂祥.物流配送车辆调度问题的模型和算法研究[D].北京: 北方交通大学, 2003.
- [7] 陈火根, 丁红钢.物流配送中心车辆调度模型及遗传算法设计[J].浙江大学学报: 工学版, 2003, 37(5): 512-516.
- [8] Baker B M, Ayeche M A.A genetic algorithm for the vehicle routing problem[J].Computers & Operations Research, 2003, 30: 787-800.
- [9] 戴树贵, 潘荫荣, 胡幼华.基于最小费用的物流配送模型及其混合单亲遗传算法[J].计算机应用, 2005, 25(11): 2681-2684.

(上接 213 页)

申请模式却可以保持各路图像的原有尺寸, 终端用户将可以获得较高的图像质量。

表1 混图模式下性能测试结果

终端个数	CPU 占用率/%	内存占用/KB
1	9.9	10 928
2	10.1	11 288
3	19.4	12 180
4	24.7	13 032

表2 用户申请模式下性能测试结果

终端个数	CPU 占用率/%	内存占用/KB
1	3.1	10 740
2	8.8	10 976
3	10.4	11 584
4	15.0	12 316

5 结论及将来的工作

本文通过采用扩展 H.225 信令、对多路视频流进行传输和显示, 实现了用户申请模式。该工作模式是对 H.323 标准现有工作模式的有益补充。实验表明, 具有用户申请模式的设备具

有良好的兼容性, 可以与传统的 H.323 设备互连互通。与混图模式相比, 用户申请模式在 CPU 和内存占用方面均具有优势, 有利于提高 MCU 的容量。用户申请模式提高了系统的交互性, 可以保持视频图像的原始尺寸。

将来的研究工作将重点围绕在 MCU 级联方式下, 实现上述用户申请模式, 为大规模应用奠定基础; 同时, 通过建立网络模型的方法^[6]来实现对 MCU 的性能进行估计和测定。

(收稿日期: 2006 年 11 月)

参考文献:

- [1] ITU-T H.323 v4 Packet-based multimedia communications systems Recommendation[S], 2000.
- [2] RFC1889 RTP A transport protocol for real-time applications[S], 1996.
- [3] ITU-T H.225.0 v5 Call signalling protocols and media stream packetization for packet-based multimedia communication systems[S], 2003.
- [4] ITU-T Q.931 ISDN user-network interface layer 3 specification for basic call control[S], 1998.
- [5] <http://www.openh323.org>.
- [6] 边学工, 胡瑞敏.基于分层排队网络模型的 MCU 性能预测及优化研究[J].计算机学报 2004, 27(2): 209-215.