

文章编号: 1001-0920(2004)07-0751-05

## 基于累积竞争神经网络的多约束路由算法

董继扬, 张军英

(西安电子科技大学 雷达信号处理国家重点实验室, 陕西 西安 710071)

**摘要:** 针对无约束最优路径问题, 提出累积竞争神经网络模型及其搜索算法, 该算法具有高度并行性, 能获得最优解、结构简单等特点。以QoS路由选择为例, 将算法推广到多约束路由问题。实验结果表明, 对于大多数多约束QoS问题, 在与相应最短路径上节点数目相当的迭代次数内, 该算法能找到问题的满意解甚至最优解。

**关键词:** 多约束路由选择; 最短路径; 累积竞争神经网络

**中图分类号:** TN 915.01; TP183

**文献标识码:** A

## Multi-constrained routing algorithm based on accumulating competition neural networks

DONG Ji-yang, ZHANG Jun-ying

(National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an 710071, China. Correspondent: DONG Ji-yang, E-mail: jydong@xidian.edu.cn)

**Abstract:** A neural network model called accumulating competition neural network (ACNN) is proposed to solve the general shortest path problem. The ACNN algorithm has the characteristics of parallelism, simplicity and efficiency in finding the global optimal solution. Taking the QoS routing as an example, the ACNN based shortest path algorithm is generalized to solve the multi-constrained routing problem. Simulation results show that the proposed method can find the satisfactory even optimal solution within a few iterations.

**Key words:** multi-constrained routing; shortest path; accumulating competition neural networks

### 1 引言

文献[1]已证明, 具有两个以上的加性或乘性约束下路径存在问题是一个NP复杂度问题, 而 $n+1$ 个约束下路径存在问题的计算复杂度等于 $n$ 个约束下最优路径问题的计算复杂度。也就是说, 即使是单约束的最优路径问题也是NP-完全问题, 即不存在一种算法可在多项式级的时间复杂度内找到一条符合要求的路径。为满足实际应用的需要, 只能通过牺牲解的最优性来降低计算复杂度。目前常用启发式算法<sup>[2]</sup>来降低运算复杂度, 从而在满意的时间内获得近似解。

QoS路由问题近年来倍受研究者的关注, 它是一种典型的多约束路由问题。利用智能优化方法(如蚁群算法、遗传算法、神经网络)<sup>[3~5]</sup>求解QoS路由问题是近年来的一个研究方向。但这些方法参数较多, 参数的选取缺乏理论指导, 算法不能保证收敛于全局最优, 计算量仍然很大, 因此不适于大规模网络以及实时处理的要求。

本文提出一种无约束路由选择算法, 即累积竞争神经网络算法。该算法具有高度的并行性, 能使网络获得全局最优解, 且网络结构简单, 易于硬件实现, 可满足大型复杂网络以及实时处理的要求。将无

收稿日期: 2003-07-07; 修回日期: 2003-11-03

基金项目: 国家自然科学基金资助项目(60371044)。

作者简介: 董继扬(1974—), 男, 福建安溪人, 博士后, 从事智能信息处理、神经网络等研究; 张军英(1961—), 女, 江苏南通人, 教授, 博士生导师, 从事智能信息处理、基因数据处理等研究。

约束的路由算法推广到多约束路由选择问题,给出了QoS路由选择算法。实验结果表明,对于大多数多约束QoS问题,该算法经过少数几步迭代就能找到问题的满意解,算法的效率得到了大幅度的提高。

## 2 QoS路由问题描述

路由网络可用一个有向赋权图  $G = (V, E)$  表示,其中:  $V$  表示交换节点集合,  $E$  表示链路集合。节点  $V_i = (d_i, l_i, J_i)$ , 分别表示该节点的时延、丢失率和时延抖动。链路  $E_{ij} = (c_{ij}, b_{ij}, d_{ij})$ , 分别表示链路  $(i, j)$  的传输费用、带宽和时延。一个多约束QoS路由请求  $R = (s, d, \text{BandW idth}, \text{Delay}, \text{Loss}, \text{Jitter}, \text{Cost})$ , 分别表示该请求的源节点、目的节点、最小传输带宽、最大传输时延、最大信息丢失率、最大传输延时抖动和最大允许传输费用。它是一条带5个约束条件的路由请求,即寻找一条从源节点到目的节点的传输路径,并满足以下要求:

- 1) 传输路径  $P$  上所有链路延时与节点延时之和不大于一请求  $R$  的最大传输延时;
- 2) 通过传输路径  $P$  上所有节点的总信息丢失率不大于一请求  $R$  的最大信息丢失率;
- 3) 传输路径  $P$  上所有链路的传输费用之和不大于一请求  $R$  的最大传输费用;
- 4) 目的节点的延时抖动满足路由请求  $R$  的要求;
- 5) 在传输路径  $P$  上每条链路的带宽不小于一请求  $R$  所需的传输带宽。

显然,约束条件越多,传输路径  $P$  的搜索就越困难。但在搜索前可先检查目的节点的延时抖动  $J_d$ , 如果  $J_d$  不满足要求,则路由请求无解。此外,还可对网络模型作简化处理,即将不满足带宽要求的链路删除,以减少约束条件的个数。于是QoS路由问题则简化为多约束路由选择问题,即寻找一条从源节点  $s$  到目的节点  $d$  的费用最小,且满足延时1)和丢失率2)约束的传输路径  $P$ 。

## 3 累积竞争神经网络模型及无约束路由选择算法

将路由网络节点间的链路费用  $c_{ij}$  作为有向赋权图的边权  $w_{ij}$ , 则无约束路由选择问题就是最短路径问题。这方面算法很多,如Dijkstra算法<sup>[6]</sup>、Floyd算法<sup>[7]</sup>、神经网络方法<sup>[8-10]</sup>等。但这些算法或是串行的(如[6,7]),或无法保证搜索到全局最优解(如[8]),或计算量很大(如[9,10]),因此都不适于大型路由网络及实时处理的要求。

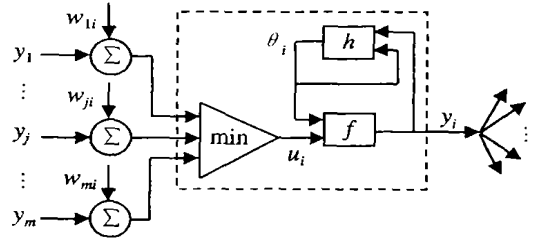


图1 累积竞争神经网络模型

针对有向赋权图最短搜索问题,本文设计出一种新的神经网络,即累积竞争神经网络。图1虚框部分表示神经网络模型,神经元  $j$  的输出  $y_j$  与连接权  $w_{ji}$  的累积值作为神经元  $i$  的一个输入,  $t$  时刻神经元  $i$  的内部激活势  $u_i(t)$  是所有输入的竞争结果,即

$$u_i(t) = \min_j \{y_j(t-1) + w_{ji}\}. \quad (1)$$

$t$  时刻神经元  $i$  的输出  $y_i$  由内部激活势  $u_i(t)$  和阈值  $\theta(t-1)$  决定,即

$$y_i(t) = f(\theta(t-1), u_i(t)) = \begin{cases} u_i(t), & u_i(t) < \theta(t-1); \\ \theta(t-1), & u_i(t) \geq \theta(t-1). \end{cases} \quad (2)$$

与一般神经网络不同,在累积竞争神经网络中,当  $y_i = 0$  时,表示神经元处于抑制状态;当  $y_i$  等于一个有限值时,表示神经元处于激活状态(又称神经元点火)。用阈值  $\theta$  记录神经元最近一次点火时的点火强度(输出值),即

$$\theta(t) = h(y_i(t), \theta(t-1)) = \begin{cases} \theta(t-1), & y_i(t) = 0; \\ y_i(t), & \text{otherwise} \end{cases} \quad (3)$$

称由上述神经元构成的网络为累积竞争神经网络(ACNN)。ACNN神经元是一种内秉的最短路径计算单元,赋权图的每个顶点对应一个神经元,边上的权重对应相应神经元间的连接权。网络按如下算法步骤运行:

### 算法1(最短树算法)

Step 1: 初始化: 设  $u_i(0) = 0, y_i(0) = 0, \theta(0) = 0, \forall i \in V$ 。

Step 2: 网络激活: 令起始神经元  $s$  的内部激活势为  $0, u_s(0) = 0$ , 使神经元  $s$  点火。

Step 3: 网络状态更新: 按式(1)~(3)并行地更新所有神经元的状态。

Step 4: 结束条件: 重复 Step 3, 直到网络不再有神经元点火为止。

Step 5: 路径回溯: 若神经元  $i$  的阈值与神经元  $j$  的阈值满足  $\theta_i = \theta_j + w_{ji}$ , 则令  $b_{ji} = 1$ ; 否则, 令  $b_{ji} = 0$ 。

= 0 路径树  $B = (b_{ij})$  就是图  $G$  中以源节点  $s$  为根的最短路树

若将神经元  $j$  的点火引起神经元  $i$  点火看作一系列自动波从神经元  $j$  传播到神经元  $i$ , 则可用自动波的传播特性<sup>[9,10]</sup> 形象地解释 ACNN 搜索有向赋权图最短树的机理:

式(1)表明: 1) 自动波从神经元  $j$  传播到神经元  $i$  时, 其强度增大了  $w_{ji}$ , 即自动波在传播过程中进行路径的累积, 自动波的强度就是该自动波所走过的路径长度; 2) 在同一时刻, 若有多列自动波传播到神经元  $i$ , 则它们发生竞争, 且强度最小的自动波获胜, 并将其强度作为神经元  $i$  的内部激活势, 而其他自动波将自行消亡

式(2)表明: 在竞争中获胜的自动波能否激活神经元  $i$  并继续向前传播, 取决于该自动波的强度是否比神经元  $i$  的阈值小 若神经元  $i$  被激活, 则神经元  $i$  输出的大小就是该自动波的强度; 若神经元  $i$  不被激活, 则其输出为无穷大(相当于自动波的传播路径在神经元  $i$  处断开).

式(3)表明: 神经元的阈值是否更新, 取决于该神经元是否被激活 若神经元被激活, 则其阈值等于神经元的输出值; 若神经元不被激活, 则其阈值不被更新 而神经元是否被激活, 取决于到达该神经元的自动波的强度是否比原阈值小, 因此, 神经元在  $t$  时刻的阈值记录了  $t$  时刻之前通过该神经元的最小强度的自动波

自动波由起始节点产生并向外传播, 自动波的传播过程就是路径的累积过程 而网络的运行过程就是自动波的传播与竞争过程, 通过竞争, 行程长的自动波及时消亡, 行程短的自动波得以生存 当网络停止运行时, 各神经元的阈值便记录了从起始节点到各自节点之间的最短路径

图 2 是有 17 个节点的无约束网络图例, 其中: 各链路上所标的值为费用值, 源节点为节点 1. 运用最短树算法, ACNN 网络经过 4 步迭代后停止 图 3 为搜索结果, 此时网络找到了以节点 1 为根的最短路树的全局最优解

以下几个性质保证了 ACNN 可搜索到最短树(证明略):

性质 1 设  $s$  为起始节点,  $D(s, i)$  表示节点  $s$  到节点  $i$  的最短路径, 则  $\theta(t+1) = \theta(i), \theta(t) = D(s, i), \forall t \geq 0$

性质 2 在运行过程中不会出现自动波传播振荡或回路现象

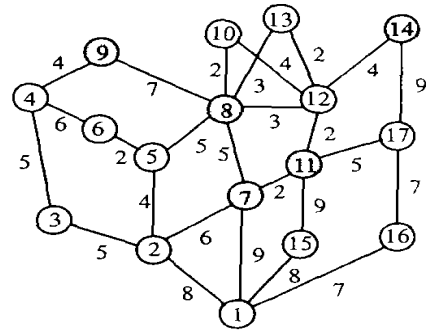


图 2 无约束路由网

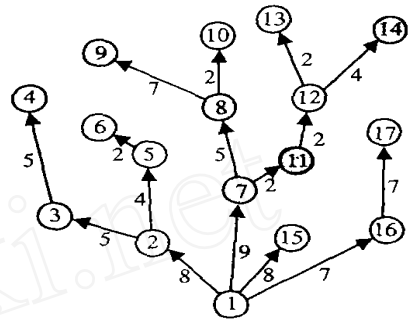


图 3 相应的最短路树

荡或回路现象

性质 3 设起始节点  $s$  到节点  $v$  的最短路径为  $p(s, v)$ ,  $D(s, v)$  为路径  $p(s, v)$  的总长度,  $N_{p(s, v)}$  为路径  $p(s, v)$  中的跃点数 则当  $t = N_{p(s, v)}$  时, 神经元  $v$  必将点火, 且  $\theta(t) = D(s, v)$ .

性质 4 网络运行结束时, 所有神经元的阈值  $\theta(t) = D(s, i)$ , 即找到了全局最优解

### 4 多约束路由选择算法

多约束路由选择问题是一个 NP- 完全问题, 即不存在一种算法可在多项式时间复杂度内找到最优解 例如 Jaffe 在 Dijkstra 算法的基础上提出了多标号算法<sup>[11]</sup>, 该算法虽能找出满足约束条件的最优解, 但其算法复杂度为  $O(n^2(m - 1)^{2n-2}c)$ . 其中:  $n$  为节点数,  $c$  为约束条件数,  $m$  为边数 其复杂度与节点数呈指数递增关系, 对于节点较多的网络, 该算法实际上是不可行的

在实际应用中, 并不一定非要找到最优解不可, 而且实验表明, 适当放宽对解的精确性要求, 往往能换来计算复杂度的大大降低 例如文献[12]从实验中观察到, 对于多标号算法, 牺牲 5% 的精确性, 其计算复杂度可降为原来的 1/3 基于这种考虑, 本文将 ACNN 算法推广到多约束路由问题

引入两个变量  $D_i(t)$  和  $L_i(t)$ , 分别表示自动波从起始节点  $s$  到节点  $i$  传播路径的总延迟和总丢失率 则有

$$Z_i^1(t) = \{j | y_j(t) + w_{ji} \text{ Cost}, \forall j \in V\}, \quad (4a)$$

$$Z_i^2(t) = \{j | L_j(t) + l_i - L_j(t) \times l_i \text{ Loss}, \forall j \in V\}, \quad (4b)$$

$$Z_i^3(t) = \{j | D_j(t) + d_{ji} + d_i \text{ Delay}, \forall j \in V\}; \quad (4c)$$

$$Z_i(t) = Z_i^1(t) \cap Z_i^2(t) \cap Z_i^3(t); \quad (5)$$

$$k = \{k | y_k(t) + w_{ki} = \min_{j \in Z_i(t)} (y_j(t) + w_{ji}), k \in Z_i(t)\}; \quad (6)$$

$$u_i(t) = y_k(t-1) + w_{ki}; \quad (7)$$

$$D_i(t) = \begin{cases} D_i(t-1), y_i(t) = 0; \\ D_k(t-1) + d_{ki} + d_i, \text{ otherwise} \end{cases} \quad (8)$$

$$L_i(t) = \begin{cases} L_i(t-1), y_i(t) = 0; \\ L_k(t-1) + l_i - L_k(t-1) \times l_i, \\ \text{otherwise} \end{cases} \quad (9)$$

其中:  $Z_i^1(t) \sim Z_i^3(t)$  分别表示满足传输费用、信息丢失率、传输时延 3 个约束条件的自动波集合;  $Z_i(t)$  表示在  $t$  时刻传播到神经元  $i$  且满足所有约束条件的自动波集合;  $k$  表示  $Z_i(t)$  中费用最小的那列自动波, 即在网络运行过程中不满足约束条件的自动波在竞争中消亡

用 ACNN 搜索多约束下的最优路径算法分为两部分: 前向搜索和逆向搜索。前向搜索是以费用最小为目的, 搜索从源节点到网络其他节点的最优路径; 逆向搜索是以费用最小为目的, 搜索网络各节点到目的节点的最优路径

**算法 2(前向搜索算法)**

Step 1: 初始化: 设  $u_i(0) = 0, y_i(0) = 0, \Theta(0) = 0, L_i(0) = 1, D_i(0) = 0, \forall i \in V$ ;

Step 2: 网络激活: 令起始神经元  $s$  的内部激活势为 0,  $u_s(0) = 0$ , 使神经元  $s$  点火;

Step 3: 网络状态更新: 按式(2) ~ (9) 并行地更新所有神经元的状态;

Step 4: 结束条件: 重复 Step 3, 直到网络不再有神经元点火为止;

Step 5: 路径回溯: 与最短树算法相同

通过前向搜索, 可得到一组数据  $\Theta, D_i$  和  $L_i$ , 分别表示从起始节点  $s$  到相应节点最优路径的费用值、延迟和丢失率。若目的节点  $d$  的阈值  $\Theta_d$ , 则找到一条满足约束条件的近似最优路径; 若  $\Theta = \infty$ , 则需要逆向搜索。即改变连接边的方向, 令

$$w_{ij} = w_{ji}, d_{ij} = d_{ji}, \forall i, j \in V, \quad (10)$$

让目的节点  $d$  点火以激活网络, 搜索从节点  $d$  到网络其他节点的最优路径。从而得到另一组数据  $\Theta, D_i$  和  $L_i$ , 分别表示各相应节点到目的节点  $d$  最优路径的费用、丢失率和延迟。则

$$\begin{cases} \Theta + \Theta, L_i + L_i - L_i \times L_i, D_i + D_i \\ L_i + L_i - L_i \times L_i \text{ Loss}, \\ D_i + D_i \text{ Delay}, \end{cases} \quad (11)$$

分别表示路径  $P(s, i) + P(i, d)$  的费用、信息丢失率和延迟。如果满足  $\Theta + \Theta, L_i + L_i - L_i \times L_i, D_i + D_i$  则说明路径  $P(s, i) + P(i, d)$  满足约束条件。从满足约束条件的路径中选择费用最少的路径, 便可得到多约束问题的一个近似最优解

**5 实验与结果**

采用一个典型例子<sup>[3-5]</sup>, 考虑图 4 的网络拓扑结构。节点  $V_i = (d_i, l_i, J_i)$ , 分别表示该节点的时延、丢失率和时延抖动; 链路  $E_{ij} = (c_{ij}, b_{ij}, d_{ij})$ , 分别表示链路  $(i, j)$  的传输费用、带宽和时延。假定有 3 个单播路由请求(1, 6), (2, 6) 和(3, 8), 它们的 QoS 要求是 Bandwidth = 70, Delay = 8, Loss =  $10^{-5}$ , Jitter = 3

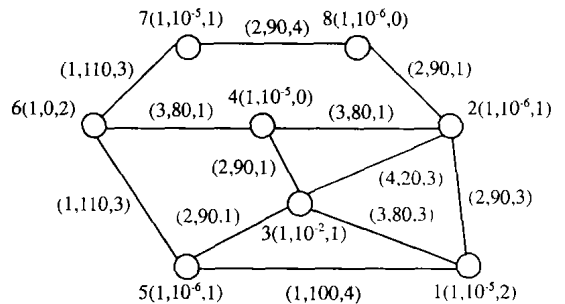


图 4 网络拓扑结构和参数

构造一个具有 8 个神经元的 ACNN 网络, 神经元间的连接权为相应节点间的链路费用, 删除不满足带宽约束的链路(2-3)。按多约束 ACNN 算法, 只需进行前向搜索便可找到 3 个请求的最优解, 其结果如表 1 所示

表 1 运用 ACNN 求解图 3 QoS 路由问题的结果

路由请求	选择的路线	费用	时延	迭代次数
(1, 6)	1 2 4 6	8	7	4
(2, 6)	2 4 6	6	3	3
(3, 8)	3 4 2 8	7	5	4

由表 1 可见, 利用 ACNN 算法, 平均只需 4 次迭代便可找到 3 个请求的最优解。本文算法的计算时间远小于蚁群算法(平均迭代 134 次)<sup>[3]</sup>和遗传算

法(平均迭代 152 次)<sup>[4]</sup>, 远远小于 Hopfield 神经网络算法(平均迭代 10 941 次)<sup>[5]</sup>.

## 6 结 语

本文针对无约束的最优路径选择问题, 提出了累积竞争神经网络模型(ACNN)及基于ACNN的最短树搜索算法. 该算法具有高度并行性, 能获得问题的全局最优解, 且ACNN的网络结构和神经元结构简单, 易于硬件实现. 以多约束QoS路由问题为例, 将ACNN的最短树算法推广到多约束的最优路径问题. 设计了相应的算法, 并给出计算机仿真结果. 仿真实验结果表明, 对于一般的多约束QoS路由选择问题, 本文算法只需少数几步迭代就能找到问题的满意解甚至最优解. 如何在不增加算法复杂度的基础上提高解的最优特性, 将是下一步的研究目标.

## 参考文献(References):

- [1] Wang Z, Crowcroft J. Quality-of-service routing for supporting multimedia applications [J]. *IEEE J of Selected Areas in Communications*, 1995, 14(7): 1219-1234.
- [2] Tedijanto T E, Onvural R O. NBBS path selection framework [J]. *IBM System J*, 1995, 34(4): 629-639.
- [3] 张素兵, 吕国英, 刘泽民, 等. 基于蚂蚁算法的QoS路由调度方法[J]. *电路与系统学报*, 2000, 5(1): 1-5 (Zhang S B, Lv G Y, Liu Z M, et al. QoS routing based on ant-algorithm [J]. *J of Circuits and Systems*, 2000, 5(1): 1-5.)
- [4] Feng X, Li J Z, Wang J V, et al. QoS routing based on genetic algorithm [J]. *Computer Communications*, 1999, 22(15-16): 1392-1399.
- [5] Ali M K M, Kanoun F. Neural networks for shortest path computation and routing in computer networks [J]. *IEEE Trans on Neural Networks*, 1993, 4(4): 941-953.
- [6] Dijkstra E W. A note on two problems in connexion with graphs [J]. *Numerical Mathematics*, 1959, 6(1): 269-271.
- [7] Floyd R. Algorithm 97: Shortest path [J]. *Communications ACM*, 1962, 6(1): 345-350.
- [8] Filipe A raujo, Bernardino Ribeiro, Luis Rodrigues. A neural network for shortest path computation [J]. *IEEE Trans on Neural Networks*, 2001, 12(5): 1067-1073.
- [9] John H Caulfield, Jason M Kinser. Finding the shortest path in the shortest time using PCNNs [J]. *IEEE Trans on Neural Networks*, 1999, 10(3): 604-606.
- [10] 张军英, 王德峰, 石美红. 输出-阈值耦合神经网络及基于此的最短路问题求解[J]. *中国科学(E 辑)*, 2003, 33(6): 522-530 (Zhang J Y, Wang D F, Shi M H. Finding shortest path in shortest time with output-threshold coupled neural networks [J]. *Science in China (Series E)*, 2003, 33(6): 522-530.)
- [11] Jaffe J M. A algorithm for finding path with multiple constraints [J]. *Networks*, 1984, 14(1): 95-116.
- [12] 刘进, 牛志升, 郑君里. 多约束ATM网络路由选择的改进多标号算法[J]. *电子学报*, 1999, 27(10): 4-7. (Liu J, Niu Z S, Zheng J L. An improved routing algorithm subject to multiple constraints for ATM networks [J]. *Acta Electronica Sinica*, 1999, 27(10): 4-7.)