

一种改进的自适应路由算法

纪竹亮, 戴连奎

(浙江大学智能系统与决策研究所, 杭州 310027)

摘 要: 在研究了自适应蚁群算法的基础上, 提出了一种改进的自适应蚁群算法, 利用计数器来协调网络拥塞和资源有效利用两者间的矛盾。仿真和分析结果表明, 该改进算法简单而有效。对于拥塞和最优路径两者在网络中的协调工作可以比较好的解决。

关键词: 动态网络; 蚁群算法; 信息素; 分布式智能; QoS

An Improved Adaptive Routing in Active Networks

Ji Zhuliang, Dai Liankui

(Institute of Intelligent Systems & Decision Making, Zhejiang University, Hangzhou 310027)

【Abstract】 This paper introduces an improved routing based on AntNet algorithm. It uses a calculator to help packets avoid congestion and at the same time use network resources effectively. And the simulation results show that the algorithm is effective and simple.

【Key words】 Active networks; AntNet algorithm; Pheromone; Distributed agents; QoS

1 蚂蚁算法^[3,8,9]

蚂蚁没有集中的管理, 个体间也不存在直接通信, 他们只是通过一种称之为信息素的东西进行间接的通信。当蚂蚁从蚁窝走到目的点进行采集食物时, 他们就会散发出信息素。信息素的停留有一个周期, 如果在他的存留周期里没有被更新, 则信息素会消失掉, 但是只要在周期内有其他的蚂蚁经过相同的路径, 则会被累加, 信息素变强。这就为以后的蚂蚁选择路径提供了条件, 因为他们总是选择信息素强的路径, 这样最后就可以达到路径的优化。在一开始的时候蚂蚁的路径选择是随机而没有规律的。只是选择了最好路径的蚂蚁它所花的时间会最短, 因此在单位时间内这条路径所累积的信息素就会比较强。这种蚂蚁寻找最短路径的方法的潜在机制是一种正反馈。即代表系统最后结果的特征被放大, 因此系统可以很快得到有用的结果。所以, 对于蚂蚁算法, 它一方面依据路径中信息素的大小来选择路径, 同时它还需要一个通用规则来实现信息素的释放以调节路径中的信息素。在这种算法中信息素的释放是基于逐跳的, 这跟蚂蚁的随机选择路径相似。但是这个算法存在以下两点主要的不足: (1) 当网络中某个路由器在坏了之后重新修好了, 这时候因为蚂蚁还是优先选择信息素强的路径走, 会使得这个路由器所在的路径没有得到很好的利用。特别是当原先这个路由所在的路径是最优路径的时候; (2) 蚂蚁总是选择信息素最强 (也就是代价最低) 的路径作为它的下一跳, 这样当流量较大的时候发生拥塞的可能性就比较大。

2 问题描述

通常可以将网络看作一个直通图 $G=(V,E)$, 其中 V 表示网络节点集, E 表示链路集。对于任意一条链路 $e \in E$, 假定其链路延迟为 $d(e)$ 。则从信源节点 s 到目的节点 t 的一条路径 $P(s,t)$ 总共所用时延为

$$D(p) = \sum_{e \in p} d(e) \quad (1)$$

本文的目标就是在均衡网络可用资源基础上寻找一条使式(1)时延最小的路径。网络中只存在一类蚂蚁, 每个路由

器中都维持着一张路由表, 因为节点的状态是动态变化的, 所以资源并不是被静态分配的, 这就要求路由表中的值也应该动态地变化。路由表的更新公式为

$$R_{i,n}^{t+1}(t+1) = \frac{R_{i,n}^{t+1}(t) + \delta}{1 + \delta} \quad (2)$$

$$R_{i,n}^{t+1}(t+1) = \frac{R_{i,n}^{t+1}(t) + \delta}{1 + \delta}, n \neq i \quad (3)$$

其中, i 表示上一节点, $i+1$ 为当前节点, s 为源节点, t 为上一周期, n 指的是所有与当前节点直接相连的节点。 δ 为定义的节点步长。定义式为: $\delta = \frac{q}{T} + b$, a, b 都是常系数, T 是蚂蚁的生命, 即从源节点到目的节点的一条路径所需时延 $\sum d_i$ 。 d_i 为路径中每条链路所用时延。 $R_{i,n}^{t+1}(t+1)$ 代表节点在新的周期中的信息素值。经过归一化后, 就可以得到类似表1这样一张路由表。示意拓扑图如图1。

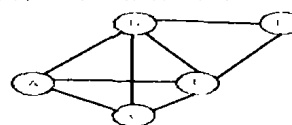


图1 示意拓扑图

表1 路由表

节点目的节点 \ A的相邻	B	C	D
D	0.16	0.21	0.63
E	0.48	0.20	0.32

归一化式子为: $\sum_n R_{i,n}^i = 1$ 。

考虑到蚁群算法所存在的问题, 在本文算法中我们引入了一个阈值 T^* , T^* 的大小由链路带宽决定, 并且使用记数

作者简介: 纪竹亮 (1977—), 男, 硕士生, 主研方向为网络控制与优化; 戴连奎, 副教授

收稿日期: 2003-04-22

E-mail: zlj@iipc.zjuedu.cn

器记录下相连节点间的信息量 T_i , 然后结合前面得到的信息素的值进行综合评价以决定路由的下一跳。为此, 定义了一个评价价值 E_i , 即

$$E_i = \sum_{j=1}^n (1 - \frac{T_i}{T}) \cdot R_{ij}, T_i \neq -1, T_i \geq 0$$

$E_i = 0$, 当 $T_i = -1$ 时。

其中 i 表示下一节点, n 为从当前节点到目的节点还有几跳, R_{ij} 为从当前节点到下一节点的信息素的值。 T_i 为计算机值。

并且定义 E_0 为 E_i 的下限值, 当 $\forall E_i \leq E_0$ 时, 信息包回传到上一节点。 $T_i = -1$ 表示路由器 i 发生故障无法正常工作。

3 算法描述

因为算法中没有中心智能节点, 智能是分布在整个网络中的。当每个数据包(本算法中可视为蚂蚁)到达当前节点(路由器)的时候, 当前节点都能够根据判断规则自主决定它的下一跳。算法的主要步骤可由以下3步组成:

(1) 初始化, 由源节点发出 m 只蚂蚁随机对路径进行探索, 得到从源节点到目的节点的所有路径并将其保存在路径集 Q 中。

(2) 遍历路径集 Q , 去除掉链路资源利用率已趋近饱和和可能存在拥塞的路径 $\{s\}$, 于是路径集变成 $Q-\{s\}$ 。

(3) 对余下的路径进行搜索, 比较各条路径的总时延, 找到时延最小的路径作为优化的结果, 同时更新此路径的信息素的值。

算法的伪代码描述算法软件实现的主要思路。

*****伪代码*****

Initialization

Compute each Links Delay;

End Initialization

for each node do

{
while (Current_time <= time_length_of_simulation)

{
if ((Current_time mod time_of_circle) == 0) then

//检查周边链路

//得到数据包的目的节点

//发出蚂蚁前往目的地

end if

for each Ant do

while(current node != destination node)

{

//判断链路是否拥塞

//更新链路状态

//信息包下一跳

//更新节点堆栈

//计数器的值加一

}//循环

end for

}

end for

*****End伪代码*****

4 算法的简单仿真和分析

本文仿真环境为PIV1.8G处理器, 256MB DDR内存, 40GB硬盘的Dell SC600服务器。网络拓扑结构图如图4所示。图2仿真了在网络中存在拥塞时信息包从源节点发往目的节点的延时情况, 并将结果跟自适应蚁群算法得到的结果进行比较。图3比较了自适应蚁群算法和本算法仿真得到的吞吐量的情况。在仿真中, 假设信息包从0节点发往15节点, 计数器的最大值 T 都取150个呼叫。仿真设定源节点每隔30s就产生一个信息包发往节点15, 步长参数 a 和 b 分别设

为0.5和2.5。实验总共仿真时间设为25min。

图2的仿真结果表明在网络拓扑结构中存在拥塞的情况下, 本文算法所用时延要比没有自适应的路由算法大, 但是一旦算法重新找到新的路径, 则延时会很快下降。

图3则仿真了本文算法与没有改进的自适应算法在吞吐量上的性能比较。从图中可用看出, 经过改进后的自适应路由算法在吞吐量上略优。这是因为它对网络中存在的拥塞具有更快的反应速度。

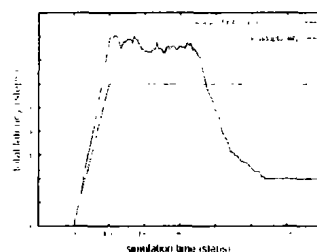


图2 时延曲线图

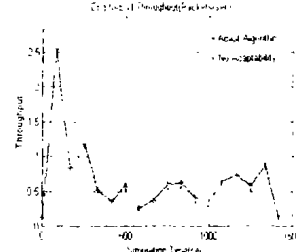


图3 吞吐量曲线图

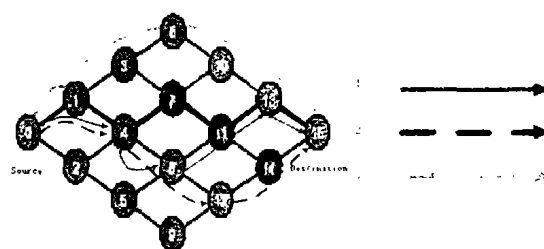


图4 信息包避开拥塞路径图

图4分析了数据包在遇到拥塞的情况下如何自主决定下一跳的情况。假设信息包从节点0发往节点15都将沿着0-1-4-7-11-13-15这样一条路径走(图中用黑粗线所示)。并且假设节点7, 11, 14都存在拥塞。当信息包到达节点4的时候, 此时节点4会察觉到节点7处发生了拥塞, 于是信息包会选择另外一条路径。这时候它选择的路径是: 4-8-11-13-15(线路1)。节点8又会发现节点11处发生拥塞, 于是重新选择路径, 这时候路径变成了4-8-12-14-15(线路2)。但是, 因为节点14也存在拥塞。这时候对于节点12、8、4都没有了前向路径可以选择, 于是他们逐一将选择路径的权利回传给上一级节点, 又回到了节点1。最终信息包找到了一条避开拥塞的最优通道: 0-1-3-6-10-13-15(线路3)。

5 结论和展望

蚁群算法的一个最主要的特征就是简单性。除了释放信息素之外, 它们无须携带任何其他的信息, 因此可以把它用于改进原有的通信协议。在本文提出的算法中, 利用计数器来协调网络拥塞和资源有效利用两者间的矛盾, 使得数据包在网络中的收发更为流畅, 网络性能也得到了较大的提高。计数器的作用其实只是在算法中增添了一个判断的依据, 从而使路径的选择更具智能化。当然, 这种算法目前还不是很成熟, 下一步的研究目标就是把它融入到流量控制算法中, 看看这样做对网络性能会有什么显著的影响。

参考文献

- Calvert K L, Zegura E W, Bhattacharjee S. Active Networking and End-to-end Arguments. Proc. Intl. Conf. Network Protocol, 1997
- Barolli L, Koyama A, Yamada T, et al. An Integrated CAC and Routing Strategy for High-speed Large-scale Networks Using Cooperative Agents. Trans. of Information Processing Society of Japan, 2001,42(2): 222-233

(下转第156页)

定系数的控制可以在短暂定位失败的情况下实现目标的持续跟踪。

定义了4种状态: 非跟踪(检测)状态, 跟踪启动状态, 不稳定跟踪状态和稳定跟踪状态。跟踪稳定系数反映了在一定时间段内定位的情况, 并作为计算预测区域的一个因素。

(1) 系统处于非跟踪状态时搜索区域为整幅图像, 跟踪的稳定系数为0;

(2) 当首次眼睛定位成功时系统进入跟踪启动状态, 并以当前帧的眼睛位置为中心进行区域膨胀得到下一帧的预测搜索区域, 由于还无法利用运动的速度和方向进行预测, 因此此时搜索区域的膨胀系数设为最大值以保证眼睛能落在预测区域内部。

(3) 当连续两帧定位成功后系统进入不稳定跟踪状态。每成功定位一帧, 提高跟踪的稳定系数直到进入稳定跟踪状态。定位失败则稳定系数下降, 当稳定系数低于某一阈值时表示跟踪目标丢失, 系统进入非跟踪状态。

(4) 一旦开始跟踪, 系统提供一个搜索区域, 搜索区域的大小和跟踪的稳定系数成反比, 跟踪稳定性越高, 搜索区域的膨胀比例越小, 反之则越大。

(5) 为了保证在定位失败的情况下仍能实现正确跟踪, 需要对原跟踪目标进行确认。在系统处于不稳定状态下眼睛被重新正确定位, 我们使用两眼距离和原跟踪目标的眼距进行比较作为确认依据, 只有变化比例符合条件才认为跟踪成功, 否则认为跟踪失败。

3 实验结果

我们在赛扬1000的机器上使用VC++6.0实现本算法, 捕获设备使用的是CS430摄像头。我们采集了150幅240 * 160的正面人脸图像进行测试, 其中100幅是在正常光照条件下(和训练肤色模型的训练样本具有相同的光照条件)捕获的, 另外50幅图像是在有色光照条件(红光)下捕获的。图6给出了一些定位的效果图。

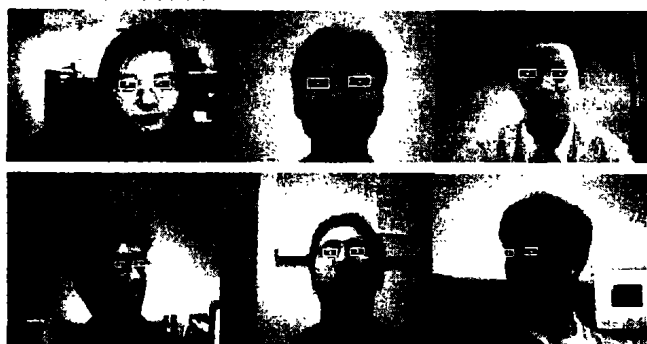


图6 定位效果图

表1是最后的实验结果。从实验结果可以看到定位准确率和光照条件有一定影响。在正常光照条件下具有很好的定位效果,有色光照条件会影响定位的准确率。对深度旋转的人脸定位,系统也有较好的鲁棒性。另外,当光照条件为侧光时,由于人脸的左右区域受光线照射情况不同,使用固定阈值提取特征点比较困难,可使用动态局部阈值来进行特征

☆☆

(上接第151页)

- 3 Oida K, Sekido M. An Agent-based Routing System for QoS Guarantees. Proc. IEEE International Conference on Systems, Man, and Cybernetics, 1999-10-12~1999-12-15:844-838
- 4 Caro G D, Dorigo M. Extending AntNet for Best Effort Quality-of-Service Routing. Proc. ANTS'98 - First International Workshop on Ant Colony Optimization, Brussels, Belgium, 1998-10-15~1998-10-16
- 5 Lipperts S, Kreller B. Mobile Agents in Telecommunications Networks — A Simulative Approach to Load Balancing. Proc. 5th Intl. Conf. Information Systems, Analysis and Synthesis, ISAS'99, 1999

—156—

点的提取。

影响定位时间的主要因素是图像大小与人脸器官特征点提取的质量,而和背景的复杂程度关系不大。在赛扬1000的机器上定位 240×160 的图像,在检测状态下需要150ms,稳定跟踪状态下只需要110ms,具有很好的实时性。

表1 最后的实验结果

图像源	样本数	错误接收		错误拒绝		检测率	正确率
		样本数	比率	样本数	比率		
正常光照图像	100	0	0%	5	5%	95%	100%
非正常光照图像	50	3	6%	8	16%	84%	92.8%
合计	150	3	2%	13	8.67%	91.33%	97.8%

4 结论和以后的工作

在本文中,我们提出了一种基于规则的快速眼睛定位算法。首先使用肤色模型和数学形态学的方法进行人脸特征点的提取,然后应用一系列规则筛选出候选眼睛对并分割出可能的眼眉区,最后通过PCA的方法计算其重建图像的信噪比进行确认。在特征提取阶段使用了肤色掩码信息,避免了复杂背景的干扰,在人脸规则阶段通过各种规则的组合,只对可信度最高的区域进行确认,大大节省了运行时间,达到了实时的目的。

同时,系统还存在需要进一步改进的地方,由于在特征点的提取阶段使用了皮肤颜色信息,因此算法会受光照条件的影响,例如在有色光照条件下(如红光)会发生定位不正确的情况,在侧光条件下,由于左右脸的灰度差异较大,在一定程度上影响了特征点提取的准确率。另外,在确认阶段受图像中人脸大小变化的影响,重建图像的信噪比也会有一定幅度的变化,在实际工作中,重建图像的信噪比阈值使用了动态更新的方法。

参考文献

- 1 Maio D, Maltoni D. Real-time Face Location on Gray-scale Static Images. *Pattern Recog.* 2000, 33:1525-1539
- 2 Propp M, Samal A. Artificial Neural Network Architecture for Human Face Detection. *Intell. Eng. Systems Artificial Neural Networks 2*, 1992: 535-540
- 3 沈兰荪, 卓力, 田栋等. 视率编码与低速率传输. 北京: 电子工业出版社, 2001-12
- 4 Hsu Rein-lien, Mohammed Abdel-Mortaleb, Jain A K. Face Detection in Color Images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2002, 24(5)
- 5 Sobottka K, Pitas I. Extraction of Facial Regions and Features Using Color and Shape Information. *International Conference on Pattern Recognition (ICPR'96)*. Vienna, Austria, 1996, III: C421-C425, 25-29
- 6 Xing Cangju, Qu Yanfeng, Wang Shoujue. Face Detection on Gray-Scale Static Image with Complex Background. *Computer-Aided Design&Computer Graphics*, 2002, 14(5)
- ☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆
- 6 Dijkstra E W. A Note on Two Problems in Connection with Graphs. *Numeric Mathematics 1*, 1959:269-271
- 7 Schoonderwoerd R, Holland O, Bruten J, et al. Antbased Load Balancing in Telecommunication Networks. *Adapt. Behav.* 5 (1996): 169-207
- 8 张素兵, 刘泽民. 基于蚂蚁算法的时延受限分布式多播路由研究. *通信学报*, 2001, 22(3)
- 9 Dorigo M, Maniezzo V, Colomni A. The Ant System: Optimization by a Colony of Cooperating Agents. Part b, *IEEE Transactions on Systems, Man and Cybernetics*, 1996, 26(1): 1-13