

doi:10.11835/j.issn.1000-582X.2013.10.013

高效的遗传蚁群组合算法在 QoS 路由上的运用

叶仕通¹, 万智萍²

(1. 广东工业大学 华立学院, 广东 增城 511325; 2. 中山大学 新华学院, 广州 510520)

摘要:针对现有遗传蚁群算法中算法融合不佳与系统易出现拥塞现象等问题,根据遗传算法与蚁群算法不同时期的优点,提出了一种高效的遗传蚁群组合算法。该算法通过根据遗传算法的群体代价关系,提出了新的融合机制;为缓解系统最优化后所产生的节点负载压力,引入了防拥塞的赏罚机制。实验结果表明,该算法能够在保证系统传输效率的同时有效的防止数据丢包现象,与传统算法相比具有高效率、低耗能以及防丢包等优势。

关键词:蚁群算法;遗传算法;拥塞;无线路由

中图分类号:TP751

文献标志码:A

文章编号:1000-582X(2013)10-082-07

An efficient combination of genetic ant colony algorithm in QoS routing

YE Shitong¹, WAN Zhiping²

(1. Huali College Guangdong University of Technology, Guangdong, Zengcheng 511325, China;
2. Xinhua College of Sun Yat-sen University, Guangzhou 510520, China)

Abstract: Existing genetic ant colony algorithm has the problem of poor integration and prone to congestion. In response to this phenomenon, this article put forward an efficient combination of genetic ant colony algorithm based on the advantages of different periods of the genetic algorithm and ant colony algorithm. The algorithm proposes a new fusion mechanism on the basis of the consideration relationship of the genetic algorithm groups. And an anti-congestion reward and punishment mechanism is introduced to alleviate the load pressure of the system nodes. The experimental results show that the algorithm can guarantee the efficiency of the system transmission, at the same time it can effectively prevent data packet loss. Compared with the traditional algorithm, it has the advantages of high efficiency, low energy and anti-dropout.

Key words: ant colony algorithm; genetic algorithm; congestion; wireless router

随着科技的发展,多媒体应用技术得以快速的发展,使得人们对路由服务质量有了更高的要求。因此,如何提高路由服务质量成为了当今无线路由领域上的研究热点。尽管近几年学者们提出了许多有关 QoS 路由算法,如基于蚁群算法的 QoS 路由算法^[1-4]、基于遗传算法的 QoS 路由算法^[5-8],但始终

无法切实满足人们的需求。由于单一算法具有一定的局限性,使得对 QoS 路由的提高具有一定的限制;如基于遗传算法的 QoS 路由算法具有大范围全局搜索的能力与潜在的并行性以及鲁棒性强的特点,但该算法在后期的搜索过程中无法有效利用前期所收集回来的信息,这使得整体搜索效率下降并

收稿日期:2013-05-11

作者简介:叶仕通(1981-),男,广东工业大学教师,主要从事计算机网路研究,(Tel)18928980165;
(E-mail)ysh888_0@126.com。

降低了算法的准确性。而基于蚁群算法的 QoS 路由算法在前期搜索过程中由于信息素的匮乏使得算法在前期搜索效率很低,但在后期搜索过程中由于积累了一定的信息素,使得算法具有较好的全局收敛能力。由此可见,如果能够将这 2 种算法结合起来,将能从本质上提高算法的搜索效率,实现算法在耗能与耗时方面的改善。但通过对蚁群遗传算法进行研究发现^[9-10],现有传统的蚁群遗传算法往往只是通过提出 2 算法的融合机制而忽略了融合过程中的拥塞现象,使得算法中融合点与融合后系统的拥塞现象得不到抑制;且大多采用简单的分段式处理,缺乏一个有效的融合机制将无法实现算法的高效融合;同时这些算法往往只是一味的提高算法的效率而忽略了算法后半段出现的节点拥塞现象。因此,文章针对以上问题,提出了一种高效遗传蚁群融合算法,通过建立一种新的蚁群算法与遗传算法融合机制,来提高算法的融合质量;同时为了避免算法过多提高搜索效率而导致节点出现拥塞现象,引入了一种防拥塞的赏罚机制,来有效利用最优解周边上的节点,实现能量的均衡分配,减少节点负载过大而导致的系统耗能过多、丢包率过高等问题。实验结果表明,其仿真结果与实验目标基本相符,具有一定的实用价值。

1 QoS 路由约束条件

一般 QoS 路由问题通常会包含多个约束条件^[11-12],如:延时 $\text{delay}(n)$ 、费用 $c(n)$ 、延时抖动 $\text{delayjitter}(n)$ 和包丢失率 $\text{packetloss}(n)$ 等。对于链路 $e \in E$,以下几种度量函数:延时 $\text{Delay}(e)$ 、费用 $\text{Cost}(e)$ 、带宽 $\text{BandWidth}(e)$ 和延时抖动 $\text{DelayJitter}(e)$ 的工作方式如下

$$\text{Cost}(p(s,d)) = \sum_{n \in p(s,d)} \text{cost}(n) + \sum_{e \in p(s,d)} \text{cost}(e), \quad (1)$$

$$\text{Band Width}(p(s,d)) = \min\{\text{BandWidth}(e), e \in p(s,d)\}, \quad (2)$$

$$\text{Delay}(p(s,d)) = \sum_{n \in p(s,d)} \text{delay}(n) + \sum_{e \in p(s,d)} \text{Delay}(e), \quad (3)$$

$$\text{DelayJitter}(p(s,d)) = \sum_{n \in p(s,d)} \text{delayjitter}(n) + \sum_{e \in p(s,d)} \text{DelayJitter}(e), \quad (4)$$

$$\text{PacketLoss}(p(s,d)) = 1 - \prod_{n \in p(s,d)} (1 - \text{packetloss}(n)). \quad (5)$$

$\text{Cost}(p(s,d))$ 为路径的总费用,即链路及结点的

费用之和; $\text{BandWidth}(p(s,d))$ 为路径的带宽为链路带宽的最小值; $\text{Delay}(p(s,d))$ 为路径的延时为链路的延时与结点的延时之和; $\text{DelayJitter}(p(s,d))$ 为路径的延时抖动为链路的延时抖动与结点的延时抖动之和; $\text{PacketLoss}(p(s,d))$ 为路径的包丢失率为结点的包丢失率之积。

为实现目标函数 F 最小化,则满足以下约束条件

- 1) 时延约束: $\text{Cost}(p(s,d)) \leq D$;
- 2) 带宽约束: $\text{BandWidth}(p(s,d)) \geq B$;
- 3) 延迟抖动约束:
 $\text{DelayJitter}(p(s,d)) \leq DJ$;
- 4) 包丢失率约束:
 $\text{PacketLoss}(p(s,d)) \leq L$;

其中: D 为平均最大时延; B 为瓶颈带宽; DJ 为最大时延抖动; L 为最大包丢失率。

2 遗传算法规则

通过结合遗传算法的特点^[13],对其进行了改进,其流程如下所示。

1) 采用的编码方式为树型编码,即树是由源节点到目的节点构成的染色体,其中遗传算法直接作用在每条染色体上。

适应度函数的定义

$$F(T_j) = F_C(A \cdot F_D + B \cdot F_B + C \cdot F_{PL}), \quad (6)$$

其中 A, B, C 分别是适应函数中时延、带宽和丢包率的比重系数。

$$F_C = \frac{1}{\text{Cost}(T_j)}; \quad (7)$$

$$F_D = \Phi(\text{Delay}(T_j) - P_d); \quad (8)$$

$$F_B = \Phi(P_b - \text{Bandwidth}(T_j)); \quad (9)$$

$$F_{PL} = \Phi(\text{PacketLoss}(T_j) - P_{pl}); \quad (10)$$

$\Phi(x) = \begin{cases} 1, & x \leq 0 \\ r, & x > 0 \end{cases}$, 其中 $\Phi(x)$ 为惩罚函数,其中惩罚程度 r 的取值范围为 0 到 1 之间。

2) 选择算子: 根据精英的优越性^[14],提出了一种精英保留策略,建立精英库,将精英库中的个体予以保留直接遗传到下一代,方便下一代的选择。其中选用轮盘赌选择机制来执行选择功能。

3) 交叉算子: 交叉算子采用单点交叉方式,即两条染色体选择一个公共节点作为交叉点,通过链路的互换,来实现该功能,但如果有多多个公共节点时,则选择第一个公共节点来进行交叉运算。

4) 变异算子: 图 1 变异节点的选择方式图中节点 w 在原始路径上经过变异,产生新的一条路径,

该路径判断最终形成新的最短路径的适用值。如下所示：

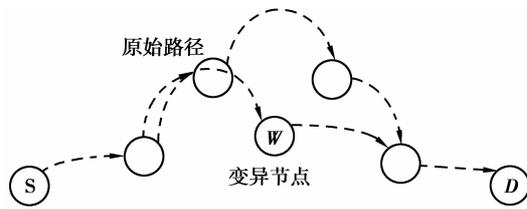


图 1 变异节点的选择方式图

其算法原则如下所示

- 1) 从原始路径的中间节点随机选取一个变异节点 w 。
- 2) 将变异节点的前驱和后继节点以适应度函数为指标, 将使适应度值最大的路径作为新的变异路径。

3 蚁群算法规则

通过结合遗传算法的特点^[15], 对其进行了分析, 如下所示

设蚁群中蚂蚁总数为 M , 其中设置 $d_{i,j}$ 为节点 i 与节点 j 之间的距离, 设 $P_{i,j}^k(t)$ 为第 k 只蚂蚁在 t 时刻由节点 i 到 j 节点的概率, 其计算公式

$$P_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,k}(t)]^\beta}{\sum_{r \in \text{allowed}_k} [\tau_{i,r}(t)]^\alpha \cdot [\eta_{i,r}(t)]^\beta}, & j \in \text{allowed}_k; \\ 0, & \text{其他,} \end{cases} \quad (11)$$

其中: α 信息启发式因子; β 望启发式因子; allowed_k 为下一可选择节点。

其中 $\eta_{i,r}(t) = 1/d_{i,r}$, 通过更新 tabu_k 来记录蚂蚁 k 当前所走过的节点最优解集合, 而为了确保能够进行路径筛选蚁群算法中还加入了信息素的挥发因子 ρ 。多路径的选优计算公式

$$r_{i,j}(t+n) = (1-\rho) \cdot \tau_{i,j}(t) + \rho \Delta r_{i,j}(t), \quad (12)$$

其中 $0 < \rho < 1$, 式中 $\Delta \tau_{i,j}(t) = \sum_{k=1}^m \Delta \tau_{i,j}^k(t)$ 的 $\Delta \tau_{i,j}^k(t)$ 表示为第 k 只蚂蚁在第 t 次迭代循环时留在路径 (i, j) 上的信息素的浓度, 而 $\Delta \tau_{i,j}(t)$ 表示为本次循环所有蚂蚁在路径 (i, j) 上所释放的信息素浓度之和。Dorigo 等人还提出了 3 种关于 $\Delta \tau_{i,j}^k(t)$ 不同模型: ant-cycle system, ant-quantity system 以及 ant-density system。其中 ant-cycle system 模型主要利用整体信息的来更新搜索, 而 ant-quantity system 与 ant-density system 主要利用局部信息来完成更

新搜索。笔者把第一种模型作为蚁群算法的更新基础模型。

遗传算法与蚁群算法问题分析: 通过对 2 种算法的研究, 发现其不同阶段, 都有其自身的搜索优势, 图 2 为蚁群算法与遗传算法的搜索曲线图, 如下所示。

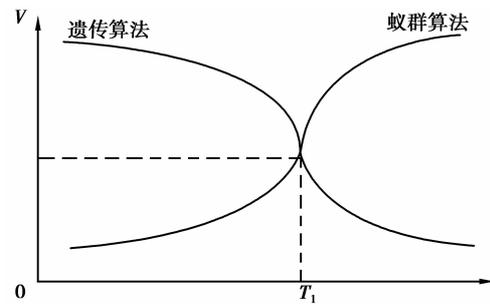


图 2 蚁群算法与遗传算法的搜索曲线图

由图 2 可以清楚的看到 2 种算法各具有自己的搜索优势。在算法搜索的前期, 遗传算法明显优于蚁群算法, 蚁群算法由于前期信息素匮乏导致搜索需要花费大量的时间对信息素进行积累以及路径的择优选择, 因此蚁群算法曲线图形在前期一直保持在搜索速度比较低的状态而蚁群算法由于后期有正反馈机制和较强的全局收敛功能, 使得算法在后期具有很好的搜索效率; 而遗传算法由于具有较好全局搜索的能力, 因此在算法的前期其搜索速度都保持在较快的状态, 但在后期由于算法缺乏对反馈信息的处理机制, 导致产生大量不必要的冗余迭代, 使得算法在后期的搜索速度反而降低了。通过分析可以发现, 如果将这 2 种算法融合在一起, 那将有效的提高 QoS 路由算法对于节点的搜索效率, 从而节约搜索所需要的时间。

4 文章算法

4.1 蚁群算法与遗传算法融合机制

通过定义遗传算法的控制函数

$$C = 1 - \frac{\overline{\text{Cost}[p(t,d)]}_m}{\overline{\text{Cost}[p(t,d)]}_{m+1}}, \quad (13)$$

其中 $\overline{\text{Cost}[p(t,d)]}_m$ 为遗传算法中第 m 次迭代后得到的群体代价的平均值, 其中 $1 < m < G$, G 为遗传算法的迭代次数, 通过 C 值的变化来动态地控制遗传算法和蚁群算法的融合时机。

通过引入 $G_{\min} \leq T \leq G_{\max}$, G_{\min} 为遗传算法最小迭代次数, G_{\max} 为遗传算法最大迭代次数, 在这个范围内, 如果连续出现几次 C 的值为 0 的情况, 则此时

可以用蚁群算法来代替遗传算法。

此时,将遗传算法所求得的解转化为蚁群算法的信息素初值, $\tau^s(i, j) = \tau^c(i, j) + \tau^g(i, j)$ 。其中 $\tau^c(i, j)$ 是根据环境给出的一个预定信息素常数, $\tau^g(i, j)$ 为遗传算法求得的路径上的信息素,当遗传算法成功转接为蚁群算法的时候,可以将遗传算法所求得的最优解集合通过公式 $\tau^g(i, j) = \frac{k}{\text{cost}[p(s, d)] \cdot \text{cost}[p(i, j)]}$ 来进行转换为信息素,其中 k 为常数,求出在最优集合中路径通过节点 i 和节点 j 的信息素叠加和最终得出新的 $\tau^g(i, j)$ 。

4.2 防拥塞赏罚机制

在算法的搜索后期,由于遗传算法已经在为蚁群算法积累了不少的信息素,使得的算法的搜索路径已经基本上确定了,又因为在蚁群算法中,蚁群对路径的选择为倾向于信息素多的一方,这就导致最优路径上的蚂蚁越积越多,原来的信息素还来不及挥发这时又有新的信息素的积累,最终超过了节点的负载能力,导致路径拥塞现象的发生为此建立了一个新的防拥塞赏罚机制,通过在搜索的过程中引入了一个新的阈值 T ,当在搜索节点的过程中,通过搜索路径与最优路径进行对比,分为 2 种情况

第 1 种节点拥塞现象:当搜索的时间过长且搜索节点超过该新的阈值 T 时,则将其认定为节点的拥塞现象,则对其引入惩罚机制。

第 2 种节点顺畅现象:此时表明有更有最优解出现的可能,为此对其进行奖励机制。

塞赏罚机制的数学模型:蚂蚁的节点选择概率计算公式如下所示。

$$P_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}(t)]^\beta}{\sum_{r \in \text{allowed}} [\tau_{i,r}(t)]^\alpha \cdot [\eta_{i,r}(t)]^\beta}, & j \in \text{allowed}_k; \\ 0, & \text{其他,} \end{cases} \quad (14)$$

$$r_{i,j}(t+1) = (1 - \rho) \cdot \tau_{i,j}(t) + \rho \sum_{k=1}^m \Delta \tau_{i,j}^k(t), \quad (15)$$

其中 $0 < \rho < 1$, 为了对体现机制的赏罚原则,则对 $\Delta \tau_{i,j}^k(t)$ 进行新的定义。

$$\Delta \tau_{i,j}^k(t) = \begin{cases} Q/n, & \text{节点顺畅;} \\ -\mu(Q/n), & \text{节点拥塞;} \\ 0, & \text{其他,} \end{cases} \quad (16)$$

根据不同的情况对其路径的信息素进行赋值。其中 n 为该搜索路径的节点, μ 为节点拥塞的赏罚系数。当搜索算法中出现节点拥塞时,对其进行惩

罚,让其他蚂蚁减少对该路径的使用,促使蚂蚁选择该路径附近符合 QoS 要求的节点,从而缓解该路径的负载压力。当搜索算法中搜索的路径比较顺畅的时候,则对其进行奖励,加快算法的搜索速度,减少不必要的搜索,从而提高算法的搜索效率。

4.3 算法设计

融合遗传-蚁群算法的混合 QoS 路由算法的主要实现步骤如下所示

步骤 1:根据 QoS 要求,删除不满足约束条件的网络节点,建立一个新的网络拓扑图。

步骤 2:初始化,设置遗传算法的控制参数。

步骤 3:随机生成初始种群

$$P(0) = (P_{1,0}, P_{2,0}, \dots, P_{i,0}, \dots, P_{m,0}) = P_0。$$

步骤 4:调整适应值:执行遗传的选择、交叉、变异等操作,直到满足遗传算法结束条件;实施精英保留策略,即将精英库中的个体予以保留直接遗传到下一代。其中遗传算法的结束条件根据蚁群算法与遗传算法的融合机制公式的 C 值来进行判断。

步骤 5:从 $P(t)$ 中选择适应值较高的若干个体,作为优化解集合。

步骤 6:对于优化解集合中的每个优化解,按初始信息素设置公式,计算资源节点对任务的信息素初值。

步骤 7:设置蚂蚁的个数为任务个数 M ,每只蚂蚁携带对应的任务信息。

步骤 8:根据概率选择公式,将蚂蚁放在资源节点 m 上,设置 $\Delta \tau_{i,j}(0) = 0$ 。

步骤 9:将各蚂蚁的初始出发点置于各自当前资源列表 $\text{tabu}(k)$ 中。

步骤 10:根据各蚂蚁所求资源列表,计算各蚂蚁所求解集的目标函数值,即每只蚂蚁所求资源列表对应的任务组的最短完成时间;记录当前的最优解判断 $K \leq M$,若还未完成搜索,则重复步骤 10 操作;否则,所有蚂蚁都成功完成自己的局部信息素更新,执行下一步骤。

步骤 11:根据步骤 10 中选出的最好解,按信息素更新公式进行各资源节点信息素的修正:若循环计数 $N \leq N_{\max}$,则执行 $N = N + 1$;在进行下一代的搜索之前,先进行路径的对比分类,引入赏罚机制如果超过阈值 T 的范围值认定节点出现拥塞现象,对其信息惩罚;如果存在最优路径,则对其进行奖励。转到下一步骤,清空所有 $\text{tabu}(k)$ 转步骤 9;否则,执行下一步骤。

步骤 12:获得最优路径,求出最优解。

其算法流程图如图 3 所示。

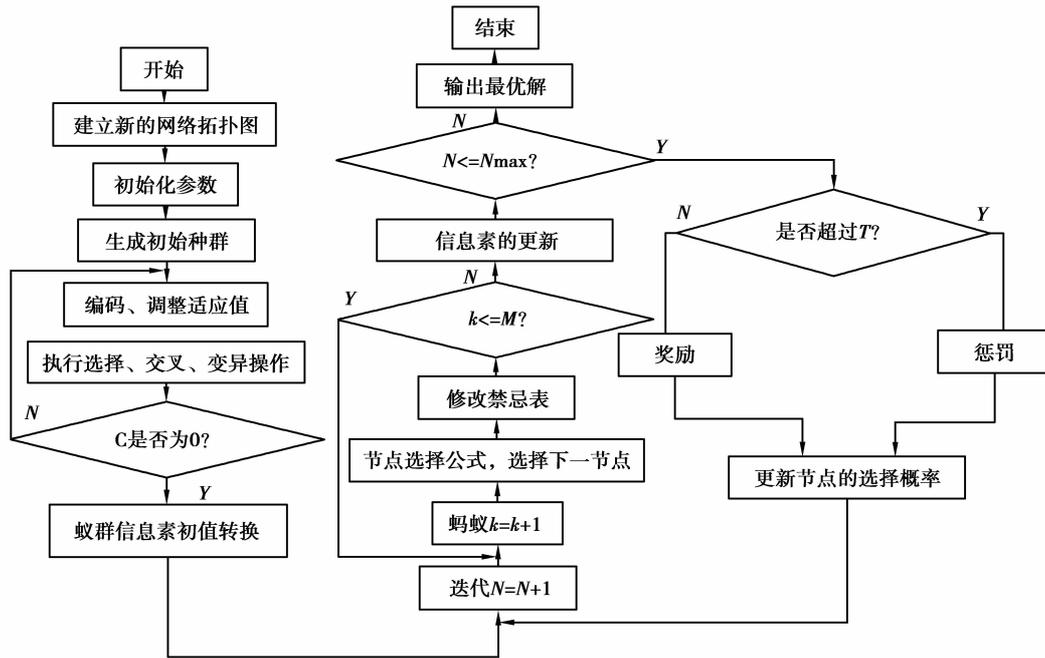


图 3 算法流程图

5 仿真与综合结果

通过 MATLAB 来对上述算法进行仿真,并将算法的性能与遗传算法和蚁群算法进行比较。实验的网络拓扑结构采用随机生成方法,建立满足 QoS 路由的约束条件的网络拓扑图如图 4 所示。

其中以节点 5 为源点、节点 14 为终点,其链路带宽在[1,10]之间、时延在[5,20]之间、代价在[2,5]之间、丢包率在[0.1,1]之间随机产生。最终得到的最优路径为图中实线链路。

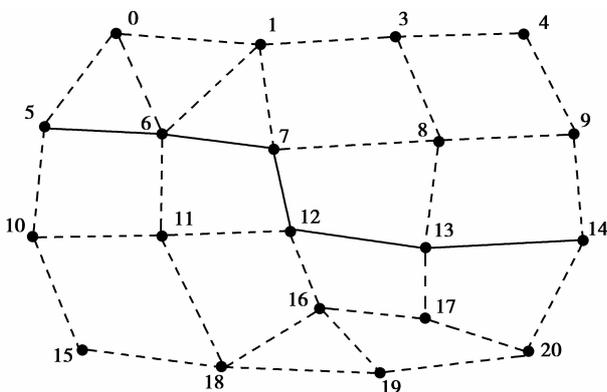


图 4 满足 QoS 路由的约束条件的网络拓扑图

为了验证算法与其他传统算法的性能差异,在 MATLAB7.10.0 的实验软件中,通过对上述几种传统的 QoS 路由算法进行性能比较,通过在相同时间内,对各算法的延时时间、包投递速率进行测试。

通过仿真实验,得表 1 为同一时间测得的各算法数据,如下所示。

表 1 同一时间测得的各算法的延时数据

时间/s	算法延迟时间/s			
	蚁群算法	遗传算法	遗传蚁群算法	笔者算法
0	0.55	0.55	0.24	0.19
50	0.68	0.62	0.31	0.18
100	0.51	0.71	0.28	0.21
150	0.63	0.65	0.35	0.21
200	0.59	0.68	0.29	0.17
250	0.58	0.72	0.27	0.22
300	0.54	0.66	0.23	0.15
350	0.51	0.67	0.29	0.21
400	0.55	0.63	0.28	0.19
450	0.61	0.69	0.28	0.20
500	0.58	0.70	0.29	0.18

从仿真结果可以看到,本算法在延时方面优于其他算法,与传统的遗传蚁群算法效果相近。为了更好的对比算法,通过 MATLAB 对传统的遗传-蚁群算法与研究算法进行仿真对比,从而检验算法是否能够实现高效防拥塞的功能。为了方便观察实验的结果,对算法的延时性以及包投递速率分别进行研究,经实验得,图 5 为传统的遗传-蚁群算法与笔者算法的对比曲线,图 6 为传统算法与笔者算法包投递速率曲线图,如下所示。

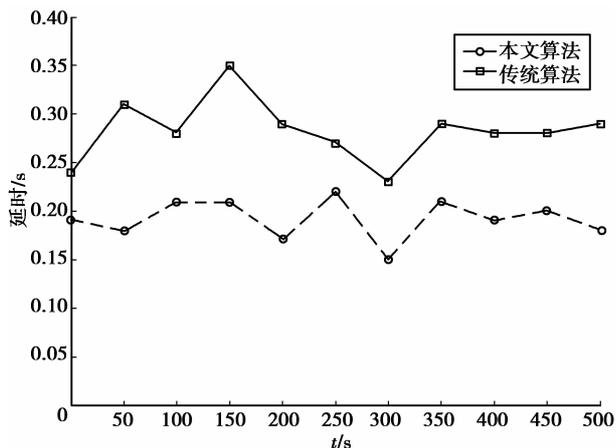


图 5 传统算法与本文算法的延时对比曲线图

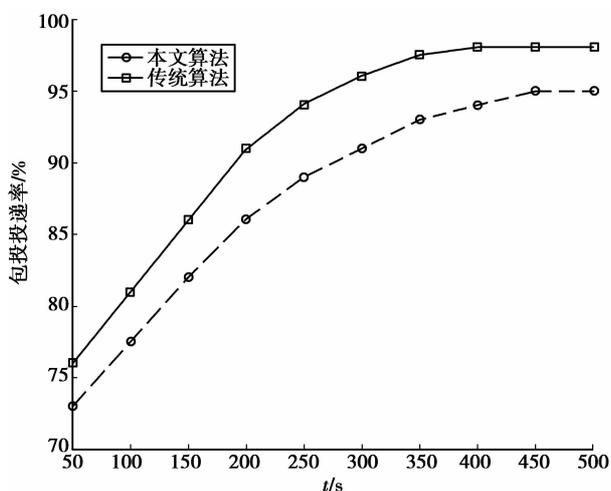


图 6 传统算法与本文算法包投递速率曲线图

由图 5 传统算法与笔者算法的延时对比曲线图可以发现:在相同的仿真条件下,笔者算法的得到的数据曲线一直保持在传统算法的下方;而在图 6 包投递速率曲线图可以发现:虽然 2 种算法的包投递速率都在持续上涨,但笔者算法的包投递速率曲线一直保持在传统算法的上方并且笔者算法的包投递速率增长幅度比较大,平均包投递速率比传统算法快 2.5%,有效的证明的笔者算法的低延时性能以及数据传输的准确性。通过对传统的遗传-蚁群算法的研究,可以发现传统的遗传-蚁群算法中缺乏一个有效的防拥塞机制,因此在算法的数据传输过程中,节点会由于传输数据的过多而出现拥塞现象,进而使数据包在传输的过程中出现丢包现象以及延迟现象,给数据的传输带来了很大的不便。而笔者在算法的后期通过引入防拥塞的赏罚机制,有效的避免了该现象的发生。经验证,实验结果与笔者预测的效果相一致,而且该算法在实验的过程中,由于做到了节点的防拥塞,使得能量得以均衡的分配到各

节点之中,有效的延长了无线网络的使用寿命,进而证明了该算法的可行性。

6 结 论

通过对蚁群算法以及遗传算法的研究与分析,提出了一种高效的遗传蚁群组合算法,该算法通过结合这 2 种算法的各个时期的搜索优势提出一种新的融合机制,有效的将这 2 种算法有机的结合在一起,到达算法优势互补的目的。同时为了避免系统中优于负载过大而造成的拥塞现象,提出了一种防拥塞的赏罚机制。实验结果与预测目标基本一致,同时能够有效抑制了局部节点的由于系统的拥塞现象,从而有利于延长了整个网络的使用寿命,在能耗方面具有一定的优势。

参考文献:

- [1] Wang H, Shi Z, Li S. Multicast routing for delay variation bound using a modified ant colony algorithm [J]. Academic Press Ltd, 2009, 01: 258-272.
- [2] Wang H, Xu H, Yi S W, et al. A tree-growth based ant colony algorithm for QoS multicast routing problem [J]. Expert Systems with Applications: An International Journal, 2011, 38 (9): 11787-11795.
- [3] Wang H, Shi Z, Ge A F, et al. An optimized ant colony algorithm based on the gradual changing orientation factor for multi-constraint QoS routing [J]. Computer Communications, 2009, 32 (4): 586-593.
- [4] Li X, Yu D T, Qin J. An improved ant colony algorithm and simulation [J]. IEEE Press. 2009:2886-2889.
- [5] Yen Y S, Chao H C, Chang R S, et al. Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs [J]. Mathematical and Computer Modelling: An International Journal. 2011, 53 (11/12): 2238-2250.
- [6] Xing H L, Liu X, Jin X, et al. A multi-granularity evolution based quantum genetic algorithm for QoS multicast routing problem in WDM networks [J]. Computer Communications, 2009, 32(2): 386-393.
- [7] Mclay L A, Goldberg D E. Efficient genetic algorithms using discretization scheduling [J]. MIT Press, 2005 (03): 353 - 385.
- [8] Leela R, Selvakumar S. Genetic algorithm approach to dynamic multiconstraint multipath QoS routing algorithm for IP networks [J]. International Journal of Communication Networks and Distributed Systems, 2010, 5(4): 392-411.
- [9] Zeng X, Tu Z Y. A study on the optimization of the constrained routing problem based on genetic-ant colony algorithm [J]. IEEE Computer Society, 2010, 218: 860-863.

- [10] Lin Z, Wang L B. Multi-Stage partner selection based on genetic-ant colony algorithm in agile supply chain network [J]. IEEE Computer Society, 2008; 1884-1889.
- [11] Liang X D, Chen M, Xiao Y, et al. MRL-CC: a novel cooperative communication protocol for QoS provisioning in wireless sensor networks [J]. International Journal of Sensor Networks, 2010, 8(2): 98-108.
- [12] Stefano B, Ugo M, Francesca R, et al. Unicast and multicast QoS routing with soft-constraint logic programming [J]. ACM Transactions on Computational Logic, 2010, 12(1): 1-48.
- [13] 王波, 王宁生. 基于遗传算法与 Tabu 搜索的拆卸序列优化算法[J]. 重庆大学学报: 自然科学版, 2006, 29(3): 23-27.
- WANG bo, WANG ningsheng. Optimizing algorithm for disassembly sequence based on tabu search and genetic algorithms [J]. Journal of Chongqing University: Noturol Science Edition, 2006, 29 (3): 23-27.
- [14] PEckart Z, PKalyanmoy D, PLothar T. Comparison of multiobjective evolutionary algorithms: empirical results[J]. MIT Press, 2000, 02 :173-195.
- [15] 陈乙雄, 吴中福, 朱郑州. 网格负载均衡策略及其蚁群优化算法[J]. 重庆大学学报, 2010, 33(10): 102-109.
- CHEN Yixiong, WU Zhongfu, ZHU Zhengzhou. Load balancing strategy and ant optimization algorithm for grids [J]. Journal of Chong qing University, 2010, 33(10): 102-109.

(编辑 侯 湘)

~~~~~

(上接第 81 页)

- [4] Frantisek P Stanislav V. Behavior protocols for software components [J]. IEEE Transactions on Software Engineering, 2002, 28(11): 1056.
- [5] Liu Y Y, Zhao Qing. The research on the construction method of evaluation indicator system based on software product line[J]. 2011 eighth international conference on fuzzy systems and knowledge discovery.
- [6] 刘紫玉. 多专业领域本体的构建及语义检索研究[D]. 北京交通大学, 2010.
- [7] Prieto-Diaz R. A faceted approach to building ontologies[J]. Proceedings of IEEE International Conference on Information Reuse and Integration(LRI 2003). 2003:458.
- [8] 李文清, 孙新, 张常有, 等. 一种本体概念的语义相似度计算方法[J]. 自动化学报, 2012(02): 229.
- LI Wenqing, SUN Xin, ZHANG Changyou, et al. A semantic similarity measure between ontological concepts[J]. Acta Automatica Sinica, 2012(02): 229.
- [9] 付青华, 李宁, 冯惠, 等. 基于刻面分类的构件检索系统的设计与实现[J]. 计算机应用与软件, 2010(06): 57.
- FU Qinghua, LIN Ning, FENG Hui, et al. Design and mplementation of component retrieval system based on faceted classification [J]. Computer Applications and Software, 2010(06): 57.
- [10] 舒远仲, 陈志勇, 彭晓红, 等. 基于刻面分类描述的构件检索方法研究[J]. 计算机工程与科学, 2010(11): 156.
- SHU Yuanzhong, CHEN Zhiyong, PENG Xiaohong, et al. A study of the component retrieving methods based on faceted classification [J]. Computer Engineering and Science, 2010(11): 156.
- [11] 徐德智, G. R. D, 陈建二. 计算本体映射纯语义查准率和查全率的框架[J]. 计算机科学, 2008(11): 128.
- XU Dezhi, G. R. D, CHEN Jianer. Framework for computing purely semantic precision and recall of ontology mapping[J]. Computer Science, 2008(11): 128.
- [12] 熊荣东. 结合 WordNet 的领域语义标注研究[D]. 重庆大学, 2011.
- [13] 王英林, 张申生. 基于本体影射规则的软件集成重构研究[J]. 计算机学报, 2001(7): 776.
- WANG Yinglin, ZHANG Shensheng. Software reconfiguration through ontology mapping[J]. Chinese Journal of Computers, 2001(7): 776.
- [14] 韦小丽, 孙涌, 张书奎, 等. 基于最大熵模型的本体概念获取方法[J]. 计算机工程, 2009(24): 114.
- WEI Xiaoli, SUN Yong, ZHANG Shukui, et al. Ontological concept extraction method based on maximum entropy model [J]. Computer Engineering, 2009(24): 114.
- [15] 徐德智, 肖文芳, 王怀民. 本体映射过程中的概念相似度计算[J]. 计算机工程与应用, 2009(09): 167.
- XU Dezhi, XIAO Wenfang, WANG Huaimin. Concept similarity calculating during the process of ontology mapping[J]. Computer Engineering and Applications, 2009(09): 167.
- [16] 王孝满, 郭成昊, 周晓明. 改进的本体概念语义相似度计算方法[J]. 指挥信息系统与技术, 2010(05): 12.
- WANG Xiaoman, GUO Chenghao, ZHOU Xiaoming. An improved measuring method of semantic similarity between ontological concepts[J]. Command Information System and Technology, 2010(05): 12.
- [17] 陈沈焰, 吴军华. 基于本体的概念语义相似度计算及其应用[J]. 微电子学与计算机, 2008(12): 96.
- CHEN Shenyan, WU Junhua. Ontology-based concept semantic similarity computation and its application[J]. Microelectronics and Computer, 2008(12): 96.

(编辑 侯 湘)