

## 卫星网络中基于多 QoS 约束的蚁群优化路由算法

魏德宾<sup>1,2</sup>, 刘 健<sup>1,2</sup>, 潘成胜<sup>1,2</sup>, 邹启杰<sup>1,2</sup>

(1. 大连大学 信息工程学院, 辽宁 大连 116622; 2. 通信与网络重点实验室, 辽宁 大连 116622)

**摘 要:** 针对蚁群算法在求解多目标优化问题时存在收敛速度慢、容易陷入局部最优解等问题, 提出一种面向卫星网络的多约束 QoS 路由算法。通过改进蚁群算法的启发函数, 将链路 QoS 信息作为蚂蚁选择下一跳节点的重要依据, 并结合排序思想与最大最小蚂蚁算法优化信息素更新规则, 获取符合当前业务的最优 QoS 路径。实验结果表明, 该算法在满足卫星网络业务多 QoS 需求的同时, 具有良好的收敛速度和寻优能力。

**关键词:** 卫星网络; 服务质量路由; 蚁群算法; 启发函数; 信息素

**中文引用格式:** 魏德宾, 刘健, 潘成胜, 等. 卫星网络中基于多 QoS 约束的蚁群优化路由算法[J]. 计算机工程, 2019, 45(7): 114-120.

**英文引用格式:** WEI Debin, LIU Jian, PAN Chengsheng, et al. Ant colony optimization routing algorithm based on multi-QoS constraints in satellite networks[J]. Computer Engineering, 2019, 45(7): 114-120.

## Ant Colony Optimization Routing Algorithm Based on Multi-QoS Constraints in Satellite Networks

WEI Debin<sup>1,2</sup>, LIU Jian<sup>1,2</sup>, PAN Chengsheng<sup>1,2</sup>, ZOU Qijie<sup>1,2</sup>

(1. College of Information Engineering, Dalian University, Dalian, Liaoning 116622, China;

2. Key Laboratory of Communication and Network, Dalian, Liaoning 116622, China)

**[Abstract]** Aiming at the problems of ant colony algorithm in solving multi-objective optimization problems, such as slow convergence speed and easy to fall into local optimal solution, a multi-constrained QoS routing algorithm for satellite networks is proposed. By improving the heuristic function of the ant colony algorithm, the link QoS information is used as an important basis for the ant to select the next hop node, and the ordering idea and the maximum and minimum ant algorithm are combined to optimize the pheromone update rule to obtain the optimal QoS path in line with the current service. Experimental results show that the algorithm has good convergence speed and optimization ability while satisfying multi-QoS requirements of satellite network services.

**[Key words]** satellite network; Quality of Service (QoS) routing; ant colony algorithm; heuristic function; pheromone

**DOI:** 10.19678/j.issn.1000-3428.0051284

### 0 概述

卫星通信系统具有覆盖范围广、接入灵活、不受地理环境和气候条件限制的特点, 现已引起人们的广泛关注。无论是地面网还是卫星网, 路由问题一直都是网络研究中的重点问题。随着全球数字化、智能化进程的加快, 卫星网络业务得到快速发展, 并且对网络传输的各项路由指标提出了更高的要求, 服务质量(Quality of Service, QoS)路由也因此受到越来越多的重视, 能为业务提供 QoS 保障的路由算法已成为卫星路由研究的热点<sup>[1-3]</sup>。

在基于 QoS 保障的路由算法中, 由于不同业务对 QoS 的需求不同, 只针对某个 QoS 参数进行优化已不能满足用户的需求, 特别是在用户的 QoS 需求不止一个且网络节点资源受限时, 多个 QoS 参数联合优化求解算法的复杂度和收敛速度是必须考虑的问题。随着智能算法的兴起, 如人工神经网络、遗传算法、粒子群算法、蚁群算法等, 由于其算法智能, 并且在求解多目标优化问题时, 表现出良好的性能, 引起了科研人员的广泛关注, 出现了较多的研究成果<sup>[4-5]</sup>。其中, 文献[6-7]提出的蚁群算法具有正反馈机制, 且鲁棒性强、分布式计算、复杂度低, 可应用

**基金项目:** 国家自然科学基金(91338104); 辽宁省自然科学基金(20170540034); 辽宁省博士科研启动基金(201601312)。

**作者简介:** 魏德宾(1978—), 男, 副教授, 主研方向为卫星网络仿真与建模; 刘 健, 硕士研究生; 潘成胜, 教授、博士、博士生导师; 邹启杰, 副教授、博士。

**收稿日期:** 2018-04-19

**修回日期:** 2018-06-04

**E-mail:** weidebin@163.com

于卫星网络来求解多约束 QoS 路由问题。文献[8]提出一种基于蚁群的 LEO 负载均衡路由算法,通过收集物理层信息做出路由决策,使用多目标优化模型来实现负载均衡,在平衡流量负载和提高报文传输速率方面表现良好,但未考虑 QoS。文献[9]提出基于跨层设计的蚁群路由算法 CL-ACR,该算法将 QoS 不敏感的业务通过高层卫星进行传输,具有搜索能力强、收敛速度快的优点,能够满足业务实时传输的要求,但其只考虑了单个目标优化,不支持多属性 QoS。文献[10]提出一种基于边界制约蚁群系统的路由算法,使用边界来限制部分路径信息素的过多积累,有效避免算法陷入早熟,具有良好的 QoS 性能,但该算法只实现了单个 QoS 指标的优化,不支持多个 QoS 指标的优化。

针对上述算法的不足,本文提出一种基于卫星网络的多 QoS 约束蚁群优化路由算法,通过优化启发函数,结合排序思想,调整链路信息素浓度。

## 1 相关概念

### 1.1 卫星网络模型

根据卫星网络的可预测性、周期性、规则性等特点,针对卫星网络拓扑动态变化的特性,本文采用基于离散化的虚拟拓扑<sup>[11]</sup>将卫星运行周期  $T$  分成  $n$  个时间片,  $T_1, T_2, \dots, T_n$ , 假设在每个时间片内,拓扑结构不变,  $n$  个时间片对应  $n$  个拓扑结构,并且由卫星存储  $n$  个不同的拓扑结构。

$G(V, E)$  表示卫星网络拓扑的基本模型,其中,  $V$  表示网络中所有卫星节点的集合,  $E = \{e(i, j) \mid i, j \in V\}$  表示连接 2 个相邻卫星节点之间的星间链路集合,并且相邻 2 个卫星节点间最多存在一条星间链路。

链路状态信息 (Link State Information, LSI) 包括链路的时延、可用带宽、丢包率、时延抖动等 QoS 信息。每个卫星节点负责维护与其直接相连的星间链路的 QoS 状态信息。

### 1.2 多约束 QoS 路由模型

QoS 路由是一种基于数据流 QoS 要求和网络可用资源相匹配的路由选择机制,是一种动态路由协议。在路径选择标准中含有可用带宽、链路和端到端路径利用率、时延、时延抖动等具体要求<sup>[12]</sup>。QoS 路由一方面要为每一个 QoS 应用请求找到满足其要求的可行路径;另一方面还要优化组合全网络的资源利用率,平衡网络负载流量,最大化网络接受其他 QoS 应用请求的能力。

在  $G(V, E)$  中,将从源节点  $s$  到目的节点  $d$  的点和边交替的序列称作一条路径,记为  $path(s, d) = v_0 e_1 v_1 e_2 \dots e_m v_m$ , 其中,  $v_0 = s, v_m = d$  且  $e_i (i = 1, 2, \dots, m)$  为路径中的链路,  $v_i (i = 1, 2, \dots, m)$  为路径经过的节点。为方便叙述,用  $e \in path(s, d)$  表示该路径中

的任意一条星间链路,用  $v \in path(s, d)$  表示该链路上的节点。将 QoS 中最具代表性和最重要的多个路径约束表示如下:

#### 1) 端到端时延

$$Delay(path) = \sum_{e \in path} Delay(e) + \sum_{v \in path} Delay(v) \quad (1)$$

其中,  $\sum_{e \in path} Delay(e)$  为路径的传播时延,  $\sum_{v \in path} Delay(v)$  为路径的排队时延。

为计算路径的传播时延,需要确定星间链路两端卫星的实际通信距离,卫星间距离关系如图 1 所示,  $R_A, R_B$  分别表示卫星 A 和 B 到地心的距离,  $L$  表示卫星 AB 间的直线距离,可通过卫星间的瞬时地心角与卫星的轨道高度计算。瞬时地心角可由式 (2) 计算得出:

$$\psi = \arccos[\sin \phi_1 \sin \phi_2 + \cos \phi_1 \cos \phi_2 \cos(\lambda_1 - \lambda_2)] \quad (2)$$

其中,  $(\lambda_1, \phi_1), (\lambda_2, \phi_2)$  为相互通信卫星的经纬度。

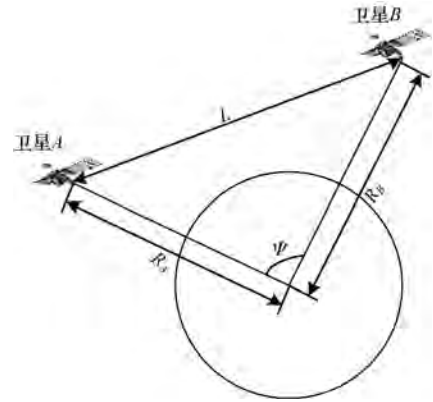


图1 卫星距离示意图

卫星间距离的计算公式为:

$$L = \sqrt{(R + H_1)^2 + (R + H_2)^2 - 2 \cos \psi (R + H_1)(R + H_2)} \quad (3)$$

其中,  $R$  为地球半径,  $H_1, H_2$  分别是卫星 A 和 B 的轨道高度。代入瞬时地心角就可以计算出两颗卫星之间的实际通信距离,通信距离  $L$  与光速  $c$  的比值即为传输时延。

节点  $v$  处关于链路  $e$  的数据包排队时延为:

$$Delay(v_e) = np(v_e) \times \frac{al}{B_{base}(e)} \quad (4)$$

其中,  $np(v_e)$  为  $v$  处关于链路  $e$  的缓冲队列中的包数,  $al$  为平均包长,  $B_{base}(e)$  为链路  $e$  的带宽。

#### 2) 通信开销

$$Cost(path) = \sum_{e \in path} Cost(e) + \sum_{v \in path} Cost(v) \quad (5)$$

通信开销包括链路的传输开销和节点的信息处理开销,通过开销小的路径传输业务,可以降低网络的负载程度。

## 3) 时延抖动

$$D\_Jitter(path) = \sum_{e \in path} D\_Jitter(e) + \sum_{v \in path} D\_Jitter(v) \quad (6)$$

时延抖动是时延的变化率,其中,  $\sum_{e \in path} D\_Jitter(e)$  为路径中由于链路拥塞或切换引起的传播时延抖动,  $\sum_{v \in path} D\_Jitter(v)$  为路径中的每个节点的排队时延抖动。

上述均属于可加性 QoS 指标,即总的 QoS 指标等于路径中各链路和节点的指标总和。

## 4) 剩余带宽

$$BandWidth(path) = \min_{e \in path} \{B_{base}(e) - B_{used}(e)\} \quad (7)$$

其中,  $B_{base}(e)$  表示链路总带宽,  $B_{used}(e)$  表示链路已用带宽。

可用带宽属于凹性参数,即在一条通信路径中,比较路径中每一条链路的可用带宽,取最小值作为该路径的可用带宽值。

## 5) 丢包率

$$P\_loss(path) = 1 - \prod_{e \in path} (1 - P\_loss(e)) \quad (8)$$

丢包率是单位时间内传输数据包中丢失的数量占总数量的比率,属于可乘性参数,其中  $P\_loss(e)$  是单位时间内路径  $path$  中链路  $e$  的丢包率。

卫星网络多约束 QoS 路由的目标是找到一条或多条路径能满足如下要求:

$$\begin{aligned} \min F &= w_1 e^{Delay(path)} + w_2 e^{D\_jitter(path)} + w_3 e^{P\_loss(path)} \\ \text{s. t. } &Dealy(path) \leq D \\ &D\_Jitter(path) \leq DJ \\ &P\_loss(path) \leq PL \end{aligned} \quad (9)$$

本文考虑卫星网络的时延、时延抖动和丢包率 3 个 QoS 属性,考虑到这 3 个属性实际值比较小,尤其是丢包率的值在 0.02% ~ 0.3% 之间,因此在式(9)中使用自然指数。由于 3 种 QoS 属性均是值越小, QoS 特性越好,因此函数  $F$  的值越小,表示路径 QoS 性能越好。其中,  $w_i$  表示路径 QoS 属性的优先系数,不同业务对应的 QoS 属性优先系数存在差别,且  $\sum_{i=1}^n w_i = 1$ ,  $D$ 、 $DJ$ 、 $PL$  分别表示时延、时延抖动、丢包率等理想的路径 QoS 属性约束值。

## 1.3 蚁群算法描述

蚁群算法<sup>[13]</sup>模拟蚂蚁觅食时会在走过路径上留下信息素,路径越短,蚂蚁在此路径上来回次数越多,则信息素浓度越大,会吸引更多的蚂蚁前来,最终找到觅食最短路径。

假定蚁群中蚂蚁总数为  $m$ ,在初始条件下,网络中所有边的信息素浓度为一常数  $C$ 。定义  $p_{ij}^k(t)$  来判定蚂蚁  $k(k=1,2,\dots,m)$  在  $t$  时刻由节点  $i$  向节点  $j$  移动的概率,具体如下:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & j \in allowed_k \\ 0, & \text{其他} \end{cases} \quad (10)$$

其中,  $allowed_k$  ( $allowed_k = \{V - tabu_k\}$ ) 表示蚂蚁  $k$  下一步可以选择的节点集,并且蚂蚁每经过一个节点,就将该节点加入禁忌表  $tabu$  中,  $\tau_{ij}(t)$  表示  $t$  时刻节点  $i$  到节点  $j$  链路上的信息素浓度,并且会随时间的推移而挥发,  $\eta_{ij}(t)$  表示  $t$  时刻节点  $i$  到节点  $j$  链路上的启发度,一般  $\eta_{ij}(t) = 1/d_{ij}(t)$ ,  $d_{ij}(t)$  表示  $t$  时刻节点  $i$  与节点  $j$  之间的距离,  $s$  表示当前  $t$  时刻节点  $i$  的邻居节点,  $\alpha$ 、 $\beta$  分别反映了蚂蚁在寻路过程中信息素和启发函数的相对重要性。

当蚂蚁完成一次循环后,需要根据式(11)、式(12)对各路径上的信息素进行更新:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (11)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (12)$$

其中,  $\Delta\tau_{ij}^k$  表示第  $k$  只蚂蚁在本次循环中在节点  $i$  到节点  $j$  链路上留下的信息素量,  $\Delta\tau_{ij}$  为本次循环中节点  $i$  到节点  $j$  链路上信息素的总量,  $\rho$  表示信息素的挥发系数,并且  $\Delta\tau_{ij}^k$  由式(13)确定:

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{蚂蚁 } k \text{ 经过 } i \text{ 和 } j \text{ 间的链路} \\ 0, & \text{其他} \end{cases} \quad (13)$$

其中,  $Q$  表示信息素强度,为常数,  $L_k$  为蚂蚁  $k$  在本次循环中所走的路径长度,即路径中包含的链路长度之和。

由上述内容可知,在蚁群算法中,单个蚂蚁在寻路过程中行为简单,而算法整体上通过信息的交互又表现出高度的智能性。在寻路过程中,蚂蚁在节点处对路径信息素进行判定进而选择下一跳节点,该过程与卫星节点依据链路状态信息为业务规划 QoS 路径的过程非常相似,故将蚁群算法进行改进,应用于卫星网络上。当为业务规划 QoS 路径时,根据蚁群算法分布式计算的特点,可以使卫星节点的计算量分散到其他节点,降低节点的计算量,同时卫星只需考虑当前链路信息来决定下一跳卫星节点,避免了链路信息的全网泛洪。因此,将蚁群算法应用于卫星网络上来解决多约束 QoS 路由问题是适合且可行的。

## 2 蚁群算法在多约束 QoS 路由中的优化

传统的蚁群算法以寻找最短路径为优先目标,并且容易陷入局部最优解。本文将多约束 QoS 路由与蚁群算法相结合,在寻路过程中充分考虑链路 QoS 属性信息,为业务提供 QoS 保障。

## 2.1 启发函数的优化

传统蚁群算法以节点之间距离的倒数作为启发函数来寻找最短路径,不考虑 QoS 信息,直接应用于卫星网络会使大量业务集中在最短路径从而造成链路拥塞,为了能支持星上业务的多 QoS 需求,现对启

发函数作如下优化: 针对星上网络业务对 QoS 的需求, 设定星间链路存在能满足业务需求的理想 QoS 属性信息。用  $x^*(i, j) = (x_1^*(i, j), x_2^*(i, j), \dots, x_n^*(i, j))$  表示归一化后的节点  $i$  到节点  $j$  链路上  $n$  个理想 QoS 属性信息, 用  $x(i, j) = (x_1(i, j), x_2(i, j), \dots, x_n(i, j))$  表示归一化后的链路实际的 QoS 属性信息。

考虑到星上业务的多样性, 不同的业务对于 QoS 需求存在差异, 因此定义节点  $i$  到节点  $j$  链路上实际属性信息与理想属性信息的 QoS 距离:

$$d(i, j) = \sqrt{\sum_{k=1}^n w_k [x_k(i, j) - x_k^*(i, j)]^2} \quad (14)$$

其中,  $w_k$  表示第  $k$  个 QoS 的权重, 满足  $\sum_{i=1}^n w_i = 1$ , 可用本征向量法<sup>[14]</sup>确定。

**定义 1 (链路优度)** 将节点  $i$  到节点  $j$  实际链路与理想链路的 QoS 距离的倒数定义为链路优度。

$$r_{ij} = \frac{1}{d(i, j)} \quad (15)$$

若实际链路与理想链路的 QoS 距离越小, 则链路优度越大, 此链路的 QoS 属性越好, 将此链路优度替换原先的短距离优先规则, 则式 (10) 变更如下:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) r_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) r_{is}^\beta(t)}, & j \in allowed_k \\ 0, & \text{其他} \end{cases} \quad (16)$$

除链路的信息素外, 链路 QoS 信息的优劣也会对蚂蚁选路造成影响, 链路 QoS 信息越好, 对蚂蚁的启发越大。

## 2.2 状态转移规则

由于蚁群算法存在停滞行为, 即通过反复的搜索, 局部最优路径的选择概率将会非常接近 100%, 因此可以通过状态转移规则来避免该问题: 结合先验知识选择与概率驱动, 如果蚂蚁  $k$  当前处于卫星节点  $i$ , 则它选择下一个卫星节点  $j$  时, 基于如下状态转移规则:

$$j = \begin{cases} \operatorname{argmax}_{j \in allowed_k} \{ \tau_{ij}^\alpha(t) r_{ij}^\beta(t) \}, & q \leq q_0 \\ J, & q > q_0 \end{cases} \quad (17)$$

其中,  $q$  是在  $[0, 1]$  区间均匀分布的随机数,  $q_0 \in [0, 1]$  为常数,  $J$  是根据式 (16) 选择的卫星节点。

上述转移规则会使蚂蚁在选路时倾向于信息素浓度大和链路优度大的边, 这样蚂蚁选路的过程就是为业务规划 QoS 路径的过程, 因此, 由式 (16) 和式 (17) 可实现蚁群算法与多 QoS 约束的结合。参数  $q_0$  可以使蚂蚁搜索其他的非最优链路, 当最优链路的 QoS 属性变差且新的最优链路出现时, 蚂蚁能较快发现并聚集, 从而避免搜索的停滞。

## 2.3 信息素更新优化

蚁群算法在寻优过程的初始阶段收敛速度较慢, 为克服这一缺陷, 将排序的思想应用到蚁群算

法中, 利用排序方法来改善蚁群算法初期的收敛速度。

**定义 2 (路径优度)** 与链路优度类似, 路径优度是源节点  $s$  到目的节点  $d$  的一条路径的属性与理想属性的 QoS 距离的倒数。

$$r_{path(s, d)} = \frac{1}{d_{path(s, d)}} \quad (18)$$

其中, 路径的 QoS 信息可根据式 (1) ~ 式 (8) 计算。

在蚁群算法寻路过程中, 当一次循环结束后, 会搜索到一定数量的可行路径, 这些路径中不一定存在全局最优解, 但路径中包含了组成全局最优解的链路。为了使信息素能快速集中到这些链路, 加快算法收敛速度, 对于已经找到的  $l$  条可行路径, 按照路径优度排序 ( $path_1 \geq path_2 \geq \dots \geq path_l$ ), 优度越大, 排名越靠前。对于搜索到的路径, 其信息素更新的大小由路径优度的排名确定, 排名越高, 更新的信息素值越大。这样使所有找到路径的蚂蚁在信息素更新时都能做出贡献, 并且依据路径优度不同, 使贡献各不相同。通过该更新方式, 可以使算法在运行初期时快速收敛。

在基于优化排序的蚁群算法中, 将信息素的全局更新规则式 (13) 作如下改进:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q \cdot r_{ij}}{l}, & \text{蚂蚁 } k \text{ 经过 } (i, j) \\ 0, & \text{其他} \end{cases} \quad (19)$$

其中,  $l$  为蚂蚁  $k$  在本次循环中找到路径的排名,  $r_{ij}$  为链路  $e(i, j)$  的链路优度。

## 2.4 链路信息素设定

信息素浓度的高低在很大程度上影响了蚂蚁对路径的选择: 当浓度取值过大时, 算法容易陷入局部循环; 当浓度取值过小时, 算法的正反馈作用减弱, 全局寻优能力变差。

为充分利用可行解, 结合最大最小蚂蚁算法<sup>[14]</sup>, 在每次循环完成后, 只对蚂蚁找到的路径进行信息素的更新, 同时为避免搜索的停滞 (陷入局部最优解), 将各条寻优路径上可能残留的信息素浓度限制范围为  $[\tau_{\min}, \tau_{\max}]$ 。每次循环完成后, 根据式 (20) 进行信息素更新。

$$\tau_{ij}(t+1) = \begin{cases} \tau_{\min}, & \tau_{ij} \leq \tau_{\min} \\ (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}, & \text{其他} \\ \tau_{\max}, & \tau_{ij} > \tau_{\max} \end{cases} \quad (20)$$

对于任意一个  $\tau_{ij}$ , 式 (21) 成立:

$$\lim_{t \rightarrow \infty} \tau_{ij}(t) = \tau_{ij} \leq \frac{1}{1-\rho} \cdot \frac{1}{f^*} \quad (21)$$

其中,  $f^*$  表示理论最优解的开销。在本文中, 考虑到链路的 QoS 属性对蚂蚁寻优过程的启发性, 将原先的寻路开销替换为链路实际 QoS 属性与理想 QoS 属性的 QoS 距离的倒数, 则式 (21) 变更为:

$$\lim_{t \rightarrow \infty} \tau_{ij}(t) = \tau_{ij} \leq \frac{1}{1-\rho} \cdot \frac{1}{r^*} \quad (22)$$

其中,  $r^*$  表示全局最优链路的 QoS 距离。

因此,  $\tau_{\max} = \frac{1}{1-\rho} \cdot \frac{1}{r^*}$ , 而  $\tau_{\min}$  可以选择一个较小的常数<sup>[15]</sup>。经过一定次数的循环后, 算法收敛于一条或几条路径, 若算法收敛于多条路径, 则此时可通过目标函数来评价路径, 进而选出最优解。

### 3 算法步骤

基于多 QoS 约束的卫星网络蚁群优化路由算法的具体实现步骤如下:

**步骤 1** 确定当前卫星周期时刻  $t$ , 得到当前时间片下卫星网络拓扑  $G(V, E)$ , 然后基于此网络拓扑进行搜索。

**步骤 2** 确定业务类型, 初始化参数: 网络中节点和链路的 QoS 参数, 多约束 QoS 条件的限定值, 源节点  $s$ , 目的节点  $d$ , 初始信息素浓度  $\tau(0)$ , 链路信息素浓度的最大最小值  $[\tau_{\min}, \tau_{\max}]$ , 挥发系数  $\rho$ , 信息素因子  $\alpha$  和启发因子  $\beta$ , 蚂蚁数量  $m$ , 最大循环次数  $N_{\max}$  等。

**步骤 3** 将源节点设置为蚂蚁的当前节点, 并加入禁忌表中。循环次数  $N = N + 1$ 。

**步骤 4** 蚂蚁根据状态转移规则式 (17), 选择下一跳节点, 并将选择的节点加入禁忌表中。

**步骤 5** 蚂蚁判断当前节点是否为目的节点, 若是目的节点, 则寻路成功, 否则, 执行步骤 6。

**步骤 6** 蚂蚁判断当前节点的 *allowed* 集合是否为空, 若为空, 则宣布寻路失败; 否则转到步骤 4。

**步骤 7** 将已经找到的路径根据路径优度进行排序, 并通过式 (19) 和式 (20) 进行信息素更新。

**步骤 8** 若  $N = N_{\max}$ , 则根据目标函数计算已经找到的路径, 输出最优解; 否则转到步骤 3。

### 4 仿真验证

本文仿真以 Iridium 系统为网络模型, 在 OPNET 中进行仿真验证。具体卫星网络参数如表 1 所示。

表 1 Iridium 系统参数设置

参数名称	参数值
轨道高度/km	780
轨道平面数	6
轨道内卫星数	11
轨道倾角/(°)	86.4
缝隙夹角/(°)	22
相邻轨道面夹角/(°)	31.6
相邻轨道间隔夹角/(°)	27

卫星间的链路带宽为 10 Mb/s, 上行链路和下行链路带宽为 8.5 Mb/s, 每个输出链路分配一个 4 MB 的缓冲区, 包的平均大小为 1 000 KB。随机选取

5 对源-目的节点, 仿真时间为 60 min, 节点请求次数随时间增加而增加。

#### 4.1 算法参数设置

从蚁群算法原理中可以看出, 参数的取值对算法性能具有重要的影响。以蚂蚁数量为例, 蚂蚁数量过少, 正反馈作用减弱, 数量过多, 路由开销增大, 影响网络性能, 一般认为节点数量的三分之二是最优蚂蚁数, 故蚂蚁数  $m = 40$ <sup>[15]</sup>, 结合文献[16]给出的参数范围, 在验证本文算法性能时的主要参数设置如下:  $Q = 100$ ,  $\tau_{\min} = 10$ ,  $\tau(0) = 50$ ,  $q_0 = 0.1$ 。对  $\alpha$ 、 $\beta$  和  $\rho$  的取值, 则由实验确定, 实验结果如图 2 所示。

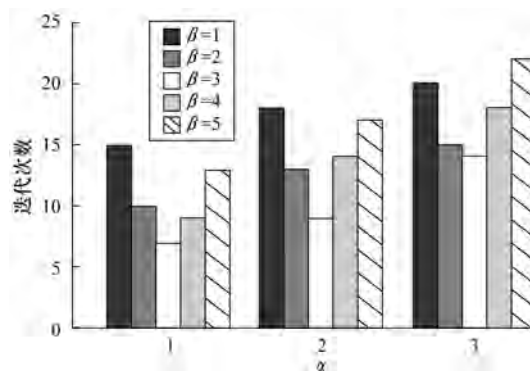


图 2  $\alpha$ 、 $\beta$  对迭代次数的影响

从多次实验可以看出, 当收敛到最优解时, 不同的  $\alpha$  和  $\beta$  取值具有不同的算法迭代次数, 随着  $\alpha$  的增大, 不论  $\beta$  取何值, 迭代次数均在增加, 因此  $\alpha = 1$ , 此时具有最小迭代次数对应的  $\beta$  值为 3, 因此  $\beta = 3$ 。在对  $\alpha$  和  $\beta$  取值进行实验时, 不论  $\rho$  取何值,  $\alpha$  和  $\beta$  对迭代次数影响的规律均相同, 图 1 是假定  $\rho = 0.6$  时的实验结果。在确定  $\alpha$  和  $\beta$  的值后,  $\rho$  取值实验如图 3 所示。

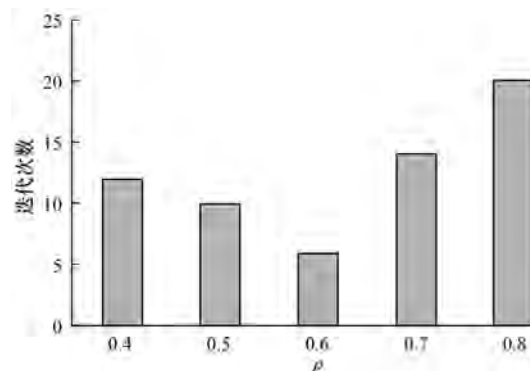


图 3  $\rho$  对迭代次数的影响

当  $\rho = 0.6$  时, 算法达到最优解所需的迭代次数最少, 因此信息素挥发因子  $\rho$  取值为 0.6。在仿真中, 考虑业务的 3 种 QoS 需求: 时延, 时延抖动和丢包率, 分别对应时延敏感(实时通话类业务)、时延抖动敏感(实时视频类业务)和丢包率敏感(下载类业务)3 类业务, 其相应的理想路径 QoS 信息如表 2 所示。

表 2 理想路径 QoS 信息

业务类型	时延/ms	时延抖动/ms	丢包率/%
实时通话类业务	60	6	0.06
实时视频类业务	67	5	0.08
下载类业务	75	8	0.03

在时间片内,卫星网络拓扑呈规则网格状,因此对所有的星间链路,设定统一的链路理想 QoS 信息:(10 ms,0.5 ms,0.01%)。由本证向量法计算出各 QoS 属性间的相对重要性,即各 QoS 属性的权重值,如表 3 所示。

表 3 相对重要性信息

业务类型	相对重要性
实时通话类业务	(0.55,0.25,0.20)
实时视频类业务	(0.28,0.58,0.14)
下载类业务	(0.30,0.16,0.54)

## 4.2 仿真结果分析

将本文提出的基于多 QoS 约束的蚁群优化路由算法与传统蚁群优化(Ant Colony Optimization, ACO)算法和基于跨层设计的蚁群路由(Cross-Layer Ant Colony Routing, CL-ACR)算法进行性能比较。

算法收敛性如图 4 所示,可以看出,随着蚂蚁数的增加,3 种算法达到最优解所需的循环次数均在减少,当蚂蚁数为 40 时,本文算法收敛性最好,在循环 6 次左右就收敛到最优解。这是因为本文算法通过优化信息素更新方式,使得所有找到路径的蚂蚁都可以对信息素进行更新,并且为所有链路设置了信息素浓度的上下界,通过这 2 种方式,加快了算法的收敛速度,因此本文算法的收敛性要优于 ACO 算法和 CL-ACR 算法。

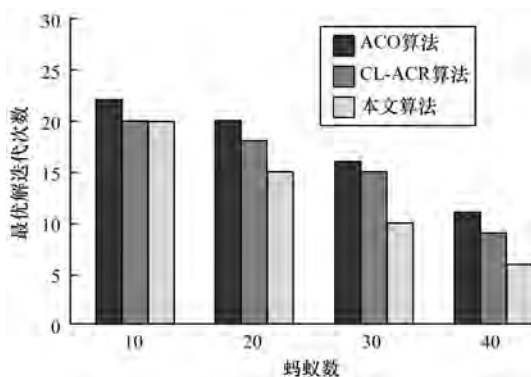


图 4 算法收敛性

网络端到端时延如图 5 所示,可以看出,在仿真初期,3 种路由算法的时延没有明显差别,随着仿真进行,本文算法在平均时延上比 CL-ACR 算法降低了 6%,比 ACO 算法降低了 11%。这是因为 ACO 算法核心是路径距离越短,对蚂蚁的启发越高,随着网络负载加重,该算法容易发生链路拥塞,造成时延增加。而本文算法在进行路径选择时,具有较低时延的路径更容易被选择,因此具有较好的时延性能。

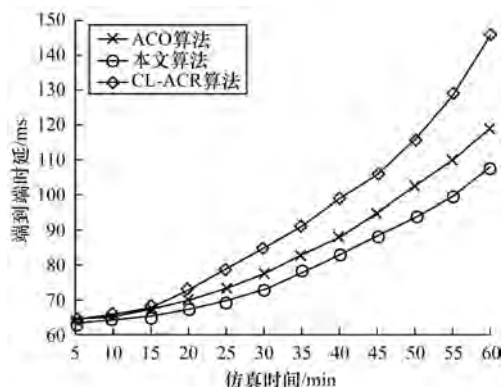


图 5 网络端到端时延

网络平均丢包率如图 6 所示,随着仿真的进行,3 种路由算法的丢包率都有所增加,但总体上本文算法丢包率比 CL-ACR 和 ACO 算法分别降低了 5% 和 8%,因为本文算法在对业务进行路径选择时,不论何种业务,均考虑到了链路的丢包信息,并且有针对性选择较低丢包率的路径进行传输。

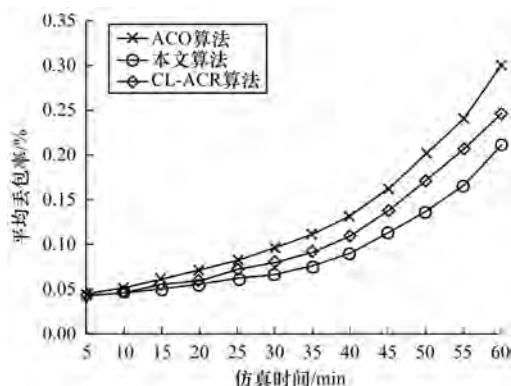


图 6 网络平均丢包率

网络平均时延抖动如图 7 所示,可以看出,本文算法在最大时延抖动上比 ACO 算法低了 9 ms,比 CL-ACR 算法低了 3 ms,并且在抖动变化率上较为平滑。这是因为 CL-ACR 路由算法中并没有考虑到时延抖动,而 ACO 算法则因为路径距离优先,使大量业务集中在特定链路上,造成了链路拥塞,导致时延抖动变化较大。

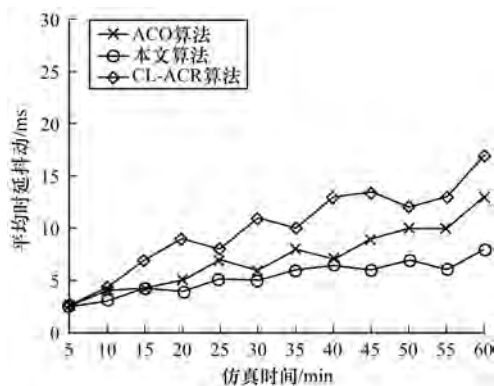


图 7 网络平均时延抖动

综上所述,本文算法通过改进传统蚁群优化算法,实现了路由的快速收敛,与现有算法相比,在 QoS 上具有更好的性能。

## 5 结束语

本文提出一种面向卫星网络的多 QoS 约束的蚁群路由算法,通过改进蚁群算法,优化启发函数,定义链路优度和路径优度,在进行路径选择时,充分考虑链路 QoS 信息,并且改善链路信息素更新方式,提高算法寻优能力和收敛速度,最终获得符合当前业务的最优 QoS 路径。仿真结果表明,本文算法较其他经典算法具有更快的收敛速度,在网络高负载情况下,QoS 性能也具有明显的优势。下一步可将多 QoS 与流量均衡相结合,在保证业务对多 QoS 需求的同时,提高卫星网络的综合利用率。

### 参考文献

- [1] 卢勇,赵有健,孙富春,等. 卫星网络路由技术[J]. 软件学报,2014,25(5):1085-1100.
- [2] WU Zhaofeng, HU Guyu, JIN Fenglin, et al. Agent-based dynamic routing in the packet-switched LEO satellite networks[C]//Proceedings of 2015 International Conference on Wireless Communications and Signal Processing. Washington D. C., USA: IEEE Press, 2015: 1-6.
- [3] GUO Andong, ZHAO Chenglin, XU Fangmin. LEO satellite routing algorithm in software defined space terrestrial integrated network [C]//Proceedings of the 17th International Symposium on Communications and Information Technologies. Washington D. C., USA: IEEE Press, 2017: 1-6.
- [4] 董明,王兴伟,黄敏. 一种面向空间信息网的 SCPS-NP QoS 单播路由机制[J]. 小型微型计算机系统, 2017, 38(7): 1425-1429.
- [5] SONG Yan, YAO Xiaomei. Design of routing protocol and node structure in wireless sensor network based on improved ant colony optimization algorithm [C]//Proceedings of 2017 International Conference on Computer Network, Electronic and Automation. Washington D. C., USA: IEEE Press, 2017: 236-240.
- [6] MOUHCINE E, KHALIFA M, MOHAMED Y. Route optimization for school bus scheduling problem based on a distributed ant colony system algorithm [C]//Proceedings of 2017 Intelligent Systems and Computer Vision. Washington D. C., USA: IEEE Press, 2017: 1-8.
- [7] WEN Guoli, ZHANG Qi, WANG Houtian, et al. An ant colony algorithm based on cross-layer design for routing and wavelength assignment in optical satellite networks[J]. China Communications, 2017, 14(8): 63-75.
- [8] WANG Houtian, ZHANG Qi, XIN Xiangjun, et al. Cross-layer design and ant-colony optimization based routing algorithm for low earth orbit satellite networks[J]. China Communications, 2013, 10(10): 37-46.
- [9] DAI Yangyang, LOU Yuansheng, LU Xin. A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing [C]//Proceedings of the 7th International Conference on Intelligent Human-Machine Systems and Cybernetics. Washington D. C., USA: IEEE Press, 2015: 428-431.
- [10] 陈莹. 基于蚁群算法的 QoS 网络路由的研究与设计[D]. 武汉: 武汉理工大学, 2010.
- [11] ZHANG Yi, ZHOU Quan, LI Jun, et al. The generation and update algorithm of routing table in satellite network[C]//Proceedings of 2015 IEEE International Conference on Communication Problem-Solving. Washington D. C., USA: IEEE Press, 2015: 619-622.
- [12] YAN Dong, GUO Jian, WANG Luyuan, et al. SADR: network status adaptive QoS dynamic routing for satellite networks [C]//Proceedings of the 13th International Conference on Signal Processing. Washington D. C., USA: IEEE Press, 2016: 1186-1190.
- [13] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 29-41.
- [14] 岳超源. 决策理论与方法[M]. 北京: 科学出版社, 2003.
- [15] STUTZLE T, HOOS H. MAX-MIN ant system and local search for the traveling salesman problem[C]//Proceedings of IEEE International Conference on Evolutionary Computation. Washington D. C., USA: IEEE Press, 1997: 309-314.
- [16] 詹士昌, 徐婕, 吴俊. 蚁群算法中有关算法参数的最优选择[J]. 科技通报, 2003, 19(5): 381-386.

编辑 陆燕菲