

基于扩展 Dijkstra 的多约束 QoS 路由研究

袁玉莹, 付 雄

(南京邮电大学 计算机学院, 江苏 南京 210003)

摘要: 为了满足以太网业务中双路径路由计算的多约束 QoS 要求, 提出了一种基于扩展 Dijkstra 的多约束 QoS 双路径路由算法, 旨在网络中能够高效地寻找到符合 QoS 约束限制的路径。算法首先对网络拓扑进行两次简化, 去除不满足带宽约束的边以及孤立的网络节点, 从而减少网络规模的复杂度以及后续的计算量; 然后利用蚁群算法的信息素更新步骤, 引入多个 QoS 度量参数的约束限制, 用是否满足约束限制条件来确定惩罚因子的数值, 通过调整目标函数值达到调整信息素增量函数大小的目的, 将时延、带宽、代价、丢包率、抖动五个 QoS 度量值约束通过目标函数转换成一个度量值约束, 最后利用扩展的 Dijkstra 算法求解满足多约束 QoS 的最优解。文中主要从迭代次数、时延、带宽、代价、丢包率、抖动六个方面对算法进行有效性分析, 实验结果表明, 该算法在时间效率以及求解精度上优于蚁群算法和遗传算法, 能很好地实现以太网业务中的多约束 QoS 双路径路由计算。

关键词: 多约束 QoS 路由; Dijkstra 算法; 蚁群算法; 时延; 带宽; 代价; 丢包率

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2021)11-0122-07

doi: 10.3969/j.issn.1673-629X.2021.11.020

Research on Multi-constrained QoS Routing Based on Extended Dijkstra

YUAN Yu-ying, FU Xiong

(School of Computer Science, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: We present a multi-constrained QoS dual-path routing algorithm based on extended Dijkstra to satisfy the requirement of finding out dual-path routing with the limit of multi-constrained QoS in Ethernet services. It is designed to figure out the path which can meet the constraints of QoS efficiently in the network. The algorithm firstly simplifies the network topology twice which can remove edges that do not meet the constraints of bandwidth and isolated network nodes, thereby reducing the complexity of network scale and the subsequent workload. Next, the multiple constraints of QoS measurement parameters is introduced by using the pheromone update step of ant colony algorithm to determine the value of the penalty factor by whether the constraint conditions are met. The value of the objective function is adjusted to achieve the purpose of adjusting the value of the pheromone increment function. The five QoS measurement constraints of delay, bandwidth, cost, loss and jitter are converted into a metric value constraint through the objective function. Finally, the optimal solution is figured out that satisfies the multi-constrained QoS by using the extended Dijkstra algorithm. We mainly analyze the effectiveness of the algorithm from six aspects of iteration, delay, bandwidth, cost, loss and jitter. Experiment shows that the proposed algorithm is better than ant colony algorithm and genetic algorithm both in terms of time efficiency and precision, and can realize multi-constrained QoS dual-path route calculation in Ethernet service.

Key words: multi-constrained QoS routing; Dijkstra algorithm; ant colony algorithm; delay; bandwidth; cost; loss

0 引言

随着 Internet 网络的广泛应用, 一方面由于实时业务对网络传输的时延、丢包率、抖动等特性较为敏感, 如果网络中发生突发性的问题, 实时业务将会受到很大的影响。另一方面, 以太网业务占用了很多的带宽, 这对当前的网络传输也提出了挑战。服务质量(QoS)

的应用范围涉及到通信网络技术、交通规划、系统工程等很多领域, 如何在网络上提供各种业务的 QoS 保证变得越来越重要。

服务质量(quality of service, QoS)是指由源节点向目的节点传输相应的业务流时, 需要满足传输层的某些性能度量约束^[1-2]。常见的 QoS 度量参数包括时

收稿日期: 2020-12-10

修回日期: 2021-04-12

基金项目: 国家自然科学基金资助项目(61202354)

作者简介: 袁玉莹(1996-), 女, 硕士生, 研究方向为移动应用安全; 付 雄, 博士, 教授, 研究方向为云计算、分布式计算。

延、带宽、代价、丢包率、抖动等,业务在网络传输过程中必须满足 QoS 约束。QoS 约束可以反映在路径约束、链路约束等方面,其中路径约束是指从源节点到目的节点的某条可行路径上,端到端的 QoS 约束,例如可乘性度量参数时延约束。链路约束指从源节点到目的节点的某条可行路径上每一条链路的约束,例如凹性度量参数带宽约束。QoS 度量参数按照性质可以分为可加性度量参数、可乘性度量参数和凹性度量参数^[3]。链路的时延、花费和抖动属于可加性度量参数,特点是 QoS 度量约束总值等于可行路径上所有链路 QoS 度量之和。链路的利用率和丢包率属于可乘性度量参数,特点是 QoS 度量约束总值等于可行路径上所有链路的 QoS 度量之积。带宽属于凹性度量参数,特点是 QoS 度量约束总值等于可行路径上所有链路的最小 QoS 度量值。

多约束 QoS 路由是根据网络中的状态信息以及不同业务对 QoS 度量参数的要求,为其选择满足多种约束(例如时延、带宽、代价、丢包率和抖动^[4-6])的可行路径的一种路由机制。QoS 约束可以是一维的参数,也可以是多维的参数,如果业务需要满足多个 QoS 度量约束时,称为多约束 QoS 路由。

目前大多数多约束 QoS 路由研究仅在网络中寻找单一的可行路径或者是单一的最优路径,如果路径上发生局部堵塞或者故障时,分组数据则会丢失,因此以太网业务中的双路径路由主要是在源节点和目的节点之间寻找两条满足多约束 QoS 条件的最优可行路径,当工作路径堵塞严重或者发生故障时,可以进行主备切换,用另一条保护路径正常进行数据的传输,在一定程度上能够消除路径上的路由振荡问题,同时减轻了网络的负载^[7]。

多约束 QoS 路由计算需要解决的问题是在源节点与目的节点之间找到能够满足多个独立 QoS 度量参数约束的最优路径。目前国际上主要的研究方法大致分为三种:精确算法(如 Dijkstra 算法, Bellman-Ford 算法)、近似算法(如拉格朗日松弛算法)以及启发式算法(如蚁群算法、遗传算法)^[8]。

对于计算多约束 QoS 路由的可行路径,当可加性约束条件或者可乘性约束条件数量在两个或两个以上的时候,该问题则是一个 NP 完全问题^[9],而研究表明传统的路由算法很难有效地解决 NP 完全^[10-11]问题。在目前所有求解从源节点至目的节点的最短路径的算法中,最为高效的是 Dijkstra 算法,它可以根据给定的任意条件,快速求解出源节点与目的节点之间的最短路径。该算法的思路是通过预处理去除不满足带宽约束的所有链路边,这样就不需要再考虑带宽的约束。除了带宽约束外,还有其他 QoS 度量参数约束时,如

果约束参数较少,可以使用 K-Shortest Path^[12]算法,将除带宽以外其他度量参数的约束条件通过某种专门设计好的函数转化为一个单独的度量参数约束^[13]。这种方法的优点是能够去除所有不能同时满足多个 QoS 约束条件的路径,但是当 QoS 度量参数约束较多时,该方法几乎不能找到一个合适的函数将多个 QoS 度量参数转化为一个度量参数。

因此针对所研究的多约束 QoS 双路径路由问题,文中设计了一种基于扩展 Dijkstra 的多约束 QoS 双路径路由算法(multi-constrained QoS dual-path routing algorithm based extended Dijkstra, ED_MCQDP),旨在网络中寻找能够满足多个 QoS 度量参数约束并且求解精度较高的双路径路由。

1 蚁群算法基本原理

蚁群算法,又称蚂蚁算法,由意大利学者 Dorigo 等人提出,是一种寻找图中最优路径的概率型算法^[14],最初应用于解决旅行商问题,目前扩展到用于解决交通路由、资源分配、通信网络中的路由问题以及负载均衡等^[15]。

算法 1:蚁群算法。

输入:节点集合 M , 蚂蚁数量 n , 迭代次数 c

输出:最优路径 path

1. $\tau_{ij}(0) = 0$ // 设置初始信息素

2. for($i = 0; i < c; i++$)

3. for($j = 1; j \leq n; j++$)

4. if($q > q_0$)

5. $p_{ij}^k(t) = \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)}{\sum_{s \in \text{allowed}_k} \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)}$ // 随机选择一个节点

6. $\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij}^k$ // 信息素局部更新

7. $\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}$ // 经过的路径信息素全局更新

8. $\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t)$ // 其他链路信息素更新

9. end if

10. end for

11. end for

12. return path

蚁群算法的选择概率公式如下:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)}{\sum_{s \in \text{allowed}_k} \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)}, & \text{若 } q > q_0 \\ 0, & \text{否则} \end{cases} \quad (1)$$

其中, $p_{ij}^k(t)$ 表示第 k 只蚂蚁在时刻 t 从节点 i 移到节点 j 的概率; q 是取 0 至 1 之间的随机数; q_0 是一个常数,是蚁群算法中状态方程的选择因子,并且满足 $0 \leq q_0 \leq 1$; $\tau_{ij}(t)$ 表示从节点 i 到节点 j 的路径在时刻 t 时信息素的强度; allowed_k 表示第 k 只蚂蚁下一跳可以

选择的城市; α 是信息启发因子, 即某条路径上的信息素对蚂蚁的影响程度; β 是期望启发式因子, 即启发信息在蚂蚁选择路径时的受重视程度; $\eta_{ij}(t)$ 是启发函数, 表示从节点 i 到节点 j 的期望程度, 表达式一般为:

$$\eta_{ij}(t) = \frac{1}{\text{cost}(i,j)} \quad (2)$$

其中, $\text{cost}(i,j)$ 表示从节点 i 到节点 j 之间路径所用的代价。对于第 k 只蚂蚁, $\text{cost}(i,j)$ 越小, 则 $\eta_{ij}(t)$ 越大, $\rho_{ij}^k(t)$ 也就越大。

蚂蚁每走一步或者完成一次遍历之后, 需要更新残留的信息素, 对旧的信息素进行一定程度的挥发, 避免累积过多的残留信息素, 导致启发信息被淹没。蚂蚁 k 从节点 i 移动到节点 j 之后, 利用公式(3)迭代更新节点 i 和节点 j 上的信息素浓度, 其中 ρ 为信息素挥发因子 (ρ 的取值范围是 $[0,1]$), 则 $1-\rho$ 是信息素残留因子。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij}^k \quad (3)$$

其中, $\Delta\tau_{ij}^k$ 为从节点 i 移动到节点 j 的路径上的信息素增量, 初始时刻 $t=0$ 时, $\tau_{ij}(0)=0$ 。 $\Delta\tau_{ij}^k$ 为第 k 只蚂蚁从节点 i 移动到节点 j 的路径上的信息素增量, 计算公式如下:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{D(i,j) \cdot L(i,j) \cdot C(i,j) \cdot J(i,j)} \\ \text{若第 } k \text{ 只蚂蚁在本次循环经过路径 } (i,j) \\ 0, \text{ 否则} \end{cases} \quad (4)$$

其中, Q 是常数, 对算法的收敛速度有一定的影响。公式(4)表示信息素增量可以根据 QoS 度量参数进行动态调整, 从而找到时延小、代价小、丢包率小以及抖动小的路径。

蚁群算法使用概率公式进行路径寻找, 发现较优解的能力很强, 但是该算法在网络规模较大时, 搜索时间很长, 而且容易陷入局部最优解, 并非全局最优解。

2 Dijkstra 算法基本原理

Dijkstra 算法的基本思想是根据路径权值的大小进行迭代, 从而计算出从源节点至目的节点的最优路径^[16]。

算法 2: Dijkstra 算法。

输入: 节点个数 n , 源节点 s , 目的节点 v

输出: 最短路径

1. $S = \{2, 3, \dots, n\}, \text{dist}[1] = 0, \text{dist}[i] = \text{weight}(1, i)$
2. while ($S.\text{length} > 0$)
3. $S = S - \{j\}$
4. if ($\text{dist}[v] + \text{weight}(v, t) < \text{dist}[t]$)
5. $\text{dist}[t] = \text{dist}[v] + \text{weight}(u, v)$

6. end while

7. return path

由算法 2 可知, Dijkstra 算法的核心步骤是从未遍历过的节点中选择路径权值最小的边, 该步骤需要遍历所有的节点得到最小权值的边, 在网络规模庞大的情况下, 搜索时间很长。此外 Dijkstra 算法是根据单个参数 (路径长度的权值) 寻找路径, 不能直接用于多个 QoS 度量参数约束的路由计算。

3 基于扩展 Dijkstra 的多约束 QoS 双路径路由算法设计

假设 $G(V, E)$ 表示网络拓扑模型, 其中 V 表示网络模型中所有网络节点 (例如交换机、路由器和主机等) 组成的集合, E 表示网络节点之间的边 (即链路) 组成的集合, 每一条边表示两个网络节点之间的直达通信路径, 每个网络节点以及每条边的 QoS 度量参数均是已知的。假设源节点为 $s (s \in V)$, 目的节点为 $d (d \in V)$ 。实际常用的 QoS 度量参数包括时延 (delay)、带宽 (bandwidth)、丢包率 (loss)、代价 (cost)、抖动 (jitter), 对于网络模型中的任意一个网络节点 $n (n \in V)$, 给定四种 QoS 度量参数, 分别为时延函数 $\text{delay}(n)$, 丢包率函数 $\text{loss}(n)$, 代价函数 $\text{cost}(n)$, 抖动函数 $\text{jitter}(n)$ 。对于网络模型中的任意一条边 $e (e \in E)$, 给定四种 QoS 度量参数, 分别为时延函数 $\text{delay}(e)$, 带宽函数 $\text{bandwidth}(e)$, 代价函数 $\text{cost}(e)$, 抖动函数 $\text{jitter}(e)$ 。

定义 1 (时延): 对于任意从源节点 s 至目的节点 d 的路由路径 $P(s, d) (s, d \in V)$, 时延 $\text{delay}(P(s, d))$ 的计算公式如下:

$$\text{delay}(P(s, d)) = \sum_{n \in P(s, d)} \text{delay}(n) + \sum_{e \in P(s, d)} \text{delay}(e) \quad (5)$$

定义 2 (带宽): 对于任意从源节点 s 至目的节点 d 的路由路径 $P(s, d) (s, d \in V)$, 带宽 $\text{bandwidth}(P(s, d))$ 的计算公式如下:

$$\text{bandwidth}(P(s, d)) = \min \{ \text{bandwidth}(e), e \in P(s, d) \} \quad (6)$$

定义 3 (丢包率): 对于任意从源节点 s 至目的节点 d 的路由路径 $P(s, d) (s, d \in V)$, 丢包率 $\text{loss}(P(s, d))$ 的计算公式如下:

$$\text{loss}(P(s, d)) = 1 - \prod_{n \in P(s, d)} (1 - \text{loss}(n)) \quad (7)$$

定义 4 (代价): 对于任意从源节点 s 至目的节点 d 的路由路径 $P(s, d) (s, d \in V)$, 代价 $\text{cost}(P(s, d))$ 的计算公式如下:

$$\text{cost}(P(s, d)) = \sum_{n \in P(s, d)} \text{cost}(n) + \sum_{e \in P(s, d)} \text{cost}(e) \quad (8)$$

定义 5(抖动):对于任意从源节点 s 至目的节点 d 的路由路径 $P(s,d)$ ($s, d \in V$), 抖动 $\text{jitter}(P(s, d))$ 的计算公式如下:

$$\text{jitter}(P(s,d)) = \sum_{n \in P(s,d)} \text{jitter}(n) + \sum_{e \in P(s,d)} \text{jitter}(e) \quad (9)$$

文中所研究的多约束 QoS 双路径路由问题即在网络拓扑图 G 中寻找从源节点 s 到目的节点 d 的两条路径, 它们均满足公式 (10) 的约束条件。

$$\begin{cases} \text{delay}(P(s,d)) \leq D \\ \text{bandwidth}(P(s,d)) \geq B \\ \text{loss}(P(s,d)) \leq L \\ \text{在满足其他的约束条件下, cost}(P(s,d)) \text{ 最小} \\ \text{jitter}(P(s,d)) \leq J \end{cases} \quad (10)$$

其中, D 、 B 、 L 、 J 分别代表以太网业务中 QoS 度量参数延迟、带宽、丢包率和抖动的约束限制。

在对网络拓扑进行路由查找之前, 先对网络拓扑结构进行精简处理, 将网络拓扑结构过滤成一个新的简单网络拓扑结构, 去除不满足带宽约束的边以及孤立的网络节点, 则算法后续不再考虑带宽度量参数的约束, 只需要考虑延迟、代价、丢包率和抖动度量参数的约束。

利用第 1 节中蚁群算法的信息素更新步骤, 引入多个 QoS 度量参数的约束限制, 惩罚因子的数值用是否满足约束限制条件来确定, 通过调整目标函数值达到调整 $\Delta\tau_{ij}^k$ 函数大小的目的, 其中 i, j 表示选择路径上相邻的两个网络节点。当 m 只蚂蚁遍历完一次之后, 计算目标函数 L_k :

$$L_k = \frac{1}{\text{cost}(P(s,d))} (AF_d + BF_1 + CF_j) \quad (11)$$

其中, $F_d = \Phi\{\text{delay}(P(n,s)) - D\}$, $F_1 = \Phi\{\text{loss}(P(n,s)) - L\}$, $F_j = \Phi\{\text{jitter}(P(n,s)) - J\}$, $\Phi_d(x) = \begin{cases} 1, & x \leq 0 \\ r_d, & x > 0 \end{cases}$, $\Phi_1(x) = \begin{cases} 1, & x \leq 0 \\ r_1, & x > 0 \end{cases}$, $\Phi_j(x) = \begin{cases} 1, & x \leq 0 \\ r_j, & x > 0 \end{cases}$, A 为延迟在目标函数中所占的比重, B 为丢包率在目标函数中所占的比重, C 为抖动在目标函数中所占的比重, 具体取值可根据不同以太网业务的需求具体设定。

算法 3: ED_MCQOP 算法。

输入: 网络拓扑图 $G(V,E)$; n 个网络节点; 源节点 s ; 目的节点 d ; 各个网络节点 (d,c,l,j) 取值; 每条链路边的 (d,b,c,j) 取值; 约束条件 (D,B,L,J) 的值。

输出: 两条符合多约束 QoS 要求的路径 p_1, p_2 。

1. remove all links which meet $\text{bandwidth}(\text{link}) < B$ and $\text{link} \in E$

2. remove all isolated nodes which belong to V

3. $P = \{s\}$

4. $N = \{\text{neighbors of } s\}$

5. for each node $v \in N$

6. $L(v) = \frac{1}{\text{cost}(P(s,v))} (AF_d + BF_1 + CF_j)$

7. $f(v) = s //$ 前一跳节点

8. end for

9. while $N \neq \emptyset$

10. $\min L = \min\{L(s,v), v \in N\}$

11. new root = i ($i \in$ the path with $\min L$ and $i \notin P$)

12. delete i from N

13. add i to P

14. for each node j ($j \in i$'s neighbors and $j \notin P$ and $j \notin N$)

15. add j to N

16. end for

17. for each node k in i 's neighbors

18. if $k \notin PN$ and $L(k) > \sum L(s,k)$

19. $L(k) = \sum L(s,k)$

20. $f(i) = k$

21. end if

22. end for

23. end while

24. if $L(s) > (AD+BL+CJ)/\text{cost}(P(s,d))$ return failure

25. else get the first path p_1

26. end if

27. for each node $v \in p_1$

28. if $(v \neq s \&\& v \neq d)$ remove node v and all links associated with node v

29. end if

30. end for

31. update P and N

32. recalculate $L(v)$

33. if $L(s) > (AD+BL+CJ)/\text{cost}(P(s,d))$ return failure

34. else get the second path p_2 and return path p_1 and p_2

35. end if

假设 $\Phi(x)$ 是 QoS 度量参数中延迟的惩罚函数, 如果路径满足延迟度量参数的约束时 ($\text{delay}(P(s, d)) \leq D$), 值为 1, 否则等于 r_d ($0 < r_d < 1$)。同理, $\Phi(x)$ 是 QoS 度量参数中丢包率的惩罚函数, 如果路径满足丢包率度量参数的约束时 ($\text{loss}(P(s, d)) \leq L$), 值为 1, 否则等于 r_1 ($0 < r_1 < 1$)。 $\Phi_j(x)$ 是 QoS 度量参数中抖动的惩罚函数, 如果路由满足抖动度量参数的约束时 ($\text{jitter}(P(s, d)) \leq J$), 值为 1, 否则等于 r_j ($0 < r_j < 1$)。 r_d, r_1, r_j 的大小分别决定了时延、丢包率和抖动的惩罚程度, 在文中实验中, 设定 $r_d = r_1 = r_j = 0.5$ 。

假设网络拓扑中有 m 只蚂蚁, n 个网络节点, 已知每个网络节点的 QoS 度量参数 d, c, l, j 取值, 以及

每条边的 QoS 度量参数 d 、 b 、 c 、 j 取值, 给定约束限制条件时延 D 、带宽 B 、代价 C 、丢包率 L 、抖动 J 的取值, ED_MCQOP 算法的伪代码如算法 3 所示。

基于扩展 Dijkstra 的多约束 QoS 双路径路由算法首先对网络拓扑图进行一系列简化, 删除不满足带宽约束的所有链路(第 1 行), 以及所有孤立的网络节点, 即没有链路链接的网络设备(第 2 行), 在网络规模较大时, 该操作可以有效地减少网络规模的复杂度, 减少算法的迭代次数。然后构造两个网络节点集合, 一个是网络节点集合 P , 集合 P 仅包含源节点 s (第 3 行), 另一个是节点集合 N , 包含源节点 s 的所有邻居节点(第 4 行), 第 5 行至第 8 行根据公式(11)引入多约束 QoS 约束限制, 计算由源节点至源节点的所有邻节点的多约束 QoS 度量值, 并存储每个节点的前一跳节点。

记录可行路径中多约束 QoS 度量值最小的路径, 将不属于节点集 P 的网络节点 i 标记为新的根节点 $root$, 并且从网络节点集 N 中删除网络节点 i , 将网络节点 i 添加到网络节点集 P 中(第 10 至第 13 行), 将新的根节点 $root$ 的所有邻节点(不包括已经在网络节点集 P 和网络节点集 N 中的网络节点)添加到网络节点集 N 中(第 14 行至第 16 行)。遍历节点 i 的所有邻节点, 进行多约束 QoS 度量值的比较更新, 核心思想与第 2 节的 Dijkstra 算法一致, 将满足条件的节点标记值进行更新, 更新节点的前一跳节点(第 17 行至第 22 行)。

当节点集 N 不为空时, 进行第 9 行至第 23 行的迭代, 得出具有最优多约束 QoS 度量值的路径, 将其与给定的多约束 QoS 度量参数值进行比较, 如果不满足给定的约束条件, 则说明网络拓扑 $G(V, E)$ 中不存在满足多约束 QoS 约束限制条件的可行路径, 计算路由失败, 算法结束(第 24 行)。否则, 得到第一条满足多约束 QoS 条件的路径 p_1 (第 25 行)。删除第一条可行路径中除了源节点与目标节点的网络节点以及每条与节点相关的链路(第 27 行至第 30 行), 该步骤是为了进一步减少网络拓扑的复杂度, 减少算法的迭代次数。然后删除网络拓扑中孤立的边和网络节点, 更新网络节点集 P 和网络节点集 N (第 31 行), 根据第 3 行至第 23 行, 重新计算网络节点的多约束 QoS 度量值 $L(v)$ (第 32 行), 将得到的最优多约束 QoS 度量值的路径与给定的多约束 QoS 度量参数值进行比较, 如果不满足给定的约束条件, 则说明当前网络拓扑中, 不存在两条满足多约束 QoS 条件的可行路径, 至此, 查找路由失败, 路由结束(第 33 行)。否则, 得到第二条满足多约束 QoS 条件的路径 p_2 , 至此, 获取到两条多约束 QoS 双路径路由, 路由路径计算成功(34 行)。

文中所设计的基于扩展 Dijkstra 的多约束 QoS 双路径路由算法, 考虑到以太网业务在实际网络中, 第一条路由由路径上的节点所在的不同子网络的规模可能非常大, 这样在某些子网络发生故障时, 进行主备切换, 使用第二条路径进行正常工作, 从而大大提高了数据传输的可靠性。同时, 本算法在寻找可行性路径的过程中, 通过网络拓扑图的精简, 减少了网络规模的复杂度, 达到了减少算法计算复杂度的目的, 并且充分考虑到了多约束 QoS 约束限制条件, 使多约束 QoS 双路径路由计算能获得最优解且整体可靠性较高。

4 实验结果与分析

本实验主要从算法的迭代次数、平均时延、平均带宽、平均代价、平均丢失率以及平均抖动来分析算法的性能。算法的迭代次数是指该算法从开始运行到最终返回满足多约束 QoS 的两条路径或者查找路由失败结束所需的迭代次数, 与算法的时间复杂度有关。在该实验中, 为了减少实验误差, 选取了 100 对源节点和目的节点, 通过重复进行实验计算平均值的方法获取算法的平均迭代次数。

实验所采用的不同规模的网络模型均由基于 MPLS 的网络管理系统通过批量建立离线设备节点生成, 网络节点数量分别设置为 100、200、300、400 和 500, 选取时延、代价、抖动三个可加性度量参数、一个可乘性度量参数丢包率以及一个凹性度量参数带宽, 由随机数生成器生成一定区间内的随机数, 在不同的网络规模下分别运用基于扩展 Dijkstra 的多约束 QoS 双路径路由算法、蚁群算法、遗传算法记录算法的迭代次数、时延、带宽、代价、丢包率以及抖动。

图 1 表示网络拓扑图中在不同的网络规模下, ED_MCQOP 算法、蚁群算法以及遗传算法的平均迭代次数。从图中可以看出, 随着网络规模的增加, 算法的平均迭代次数也随之增加。同时, 蚁群算法的平均迭代次数增长得最快, ED_MCQOP 算法的平均迭代次数在不同网络规模下相对于蚁群算法和遗传算法都较少。

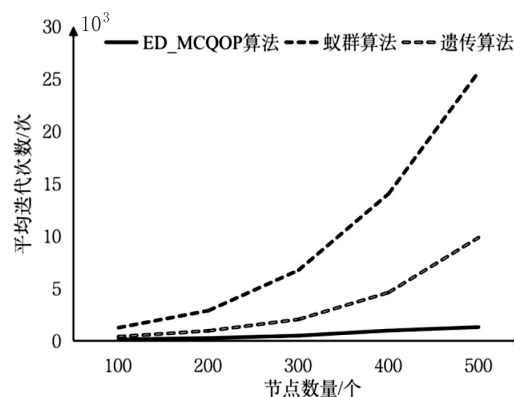


图 1 不同网络规模下三种算法的平均迭代次数

假设时延约束 $\text{delay} = 30 \text{ ms}$, 图 2 表示网络拓扑图中不同网络规模下 ED_MCQOP 算法、蚁群算法以及遗传算法的平均时延。从图中可以看出, 相较于蚁群算

法和遗传算法, ED_MCQOP 算法的平均时延大多数情况下相对较低。

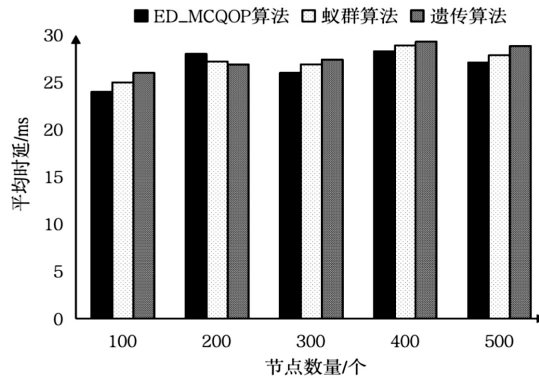


图 2 不同网络规模下三种算法的平均时延

假设带宽约束 $\text{bandwidth} = 15 \text{ Mbit/s}$, 图 3 表示网络拓扑图中不同网络规模下 ED_MCQOP 算法、蚁群算法以及遗传算法的平均带宽。从图中可以看出, 相较

于蚁群算法和遗传算法, ED_MCQOP 算法的平均带宽大多数情况下相对较低。

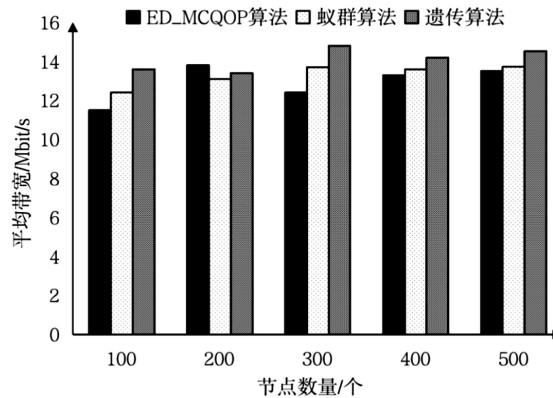


图 3 不同网络规模下三种算法的平均带宽

假设代价约束 $\text{cost} = 30$, 图 4 表示网络拓扑图中不同网络规模下 ED_MCQOP 算法、蚁群算法以及遗传算

法的平均代价。从图中可以看出, 一般情况下, ED_MCQOP 算法的平均带宽相对较低。

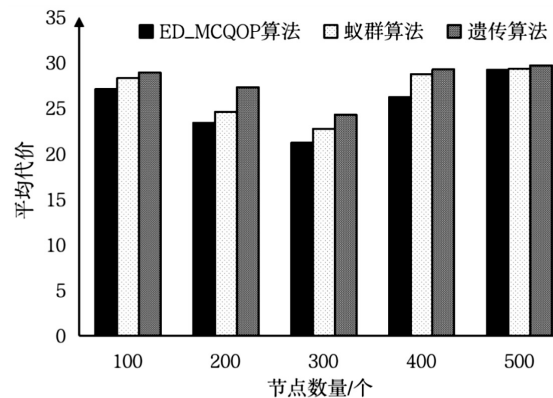


图 4 不同网络规模下三种算法的平均代价

假设丢包率约束 $\text{loss} = 4.5 \times 10^{-3} \%$, 图 5 表示网络拓扑图中不同网络规模下 ED_MCQOP 算法、蚁群算法以及遗传算法的平均丢包率。从图中可以看出, 相较于蚁群算法和遗传算法, ED_MCQOP 算法的平均丢包率大多数情况下相对较低。

假设抖动约束 $\text{jitter} = 20 \text{ ms}$, 图 6 表示网络拓扑图中不同网络规模下 ED_MCQOP 算法、蚁群算法以及遗传算法的平均抖动。从图中可以看出, 相较于蚁群算法和遗传算法, ED_MCQOP 算法的平均抖动大多数情况下相对较低。

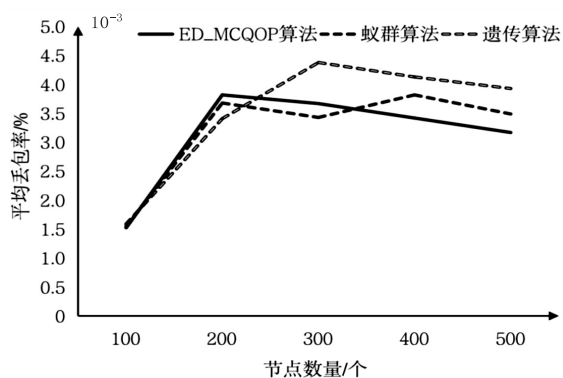


图 5 不同网络规模下三种算法的平均丢包率

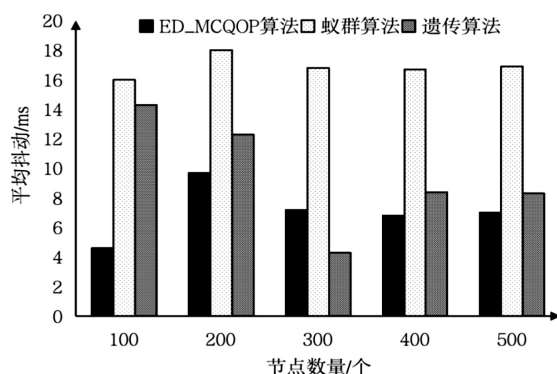


图 6 不同网络规模下三种算法的平均抖动

5 结束语

文中基于扩展的 Dijkstra 算法和蚁群算法的融合,研究包含延迟、带宽、代价、丢包率和抖动度量参数的多约束 QoS 双路路径路由问题,实验结果表明,基于扩展 Dijkstra 的多约束 QoS 双路路径路由算法在求解性能上优于蚁群算法和遗传算法,可用于求解网络拓扑中的 NP 完全问题。

参考文献:

- [1] 李新明. 基于群智能优化算法的多约束 QoS 单播路由问题研究[D]. 武汉:华中科技大学,2019.
- [2] LIANG H, WEN X, LIU Y, et al. Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning[J]. Robotics and Computer-Integrated Manufacturing,2021,67:101991.

- [3] 张继军,高 鹏. 基于分组网络的服务质量保证[M]. 北京:北京邮电大学出版社,2004:88-124.
- [4] DOSHI M, KAMDAR A. Multi-constraint QoS disjoint multipath routing in SDN[C]//Proceedings of international conference on computing, communication and signal processing. Lonere-Raigad, India:Springer,2018:377-387.
- [5] WANG X, YU S, DAI J Y, et al. A multiple constraint quality of service routing algorithm base on dominating tree[C]//IEEE international conference on computational intelligence and software engineering. Wuhan:IEEE,2009:1-4.
- [6] 孔祥彬. 软件定义联网的路由选择技术研究[D]. 南京:南京邮电大学,2017.
- [7] CHEN S, NAHRSTEDT K. An overview of quality of service routing for next-generation high-speed networks: problems and solutions[J]. IEEE Network,1998,12(6):64-79.
- [8] 冯 颖. 基于改进拉格朗日松弛的 HFS 多处理器任务调度优化研究[D]. 郑州:郑州大学,2016.
- [9] GENG H, SHI X, YIN X, et al. Algebra and algorithms for multipath QoS routing in link state networks[J]. Journal of Communications and Networks,2017,19(2):189-200.
- [10] WANG Z, CROWCROFT J. Quality-of-service routing for supporting multimedia applications[J]. IEEE Journal on Selected Areas in Communications,1996,14(7):1228-1234.
- [11] 段海滨. 蚁群算法原理及其应用[M]. 北京:科学出版社,2005:29-33.
- [12] WANG Zhengying, SHI Bingxin, ERDUN Z. Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm[J]. Computer Communications,2001,24(7-8):685-692.
- [13] 徐 涛,丁晓璐,李建伏. K 最短路径算法综述[J]. 计算机工程与设计,2013,34(11):3900-3906.
- [14] 彭 艺,李启骞,朱 豪,等. 一种基于排序蚁群算法的无线 Mesh 多路径路由协议[J]. 通信技术,2020,53(3):634-640.
- [15] 高 巍,杨亚东. 引入社会分工机制的蚁群算法[J]. 计算机光盘软件与应用,2014(17):299-300.
- [16] 刘捷超. 面向智能电网的微功率网络中的备用路径算法研究[D]. 西安:西安电子科技大学,2017.