



研究与开发

基于时延约束的广域网络 拓扑设计和容量规划

徐晓青, 唐宏, 阮科, 武娟, 刘晓军

(中国电信股份有限公司研究院, 广东 广州 510630)

摘要: 为了优化网络性能和降低建设成本, 应用线性规划和启发式贪婪取走算法, 基于时延约束对广域网络规划中拓扑设计和容量确定问题进行研究。首先, 考虑需求的权重, 采用线性规划求解给定路径下的最低时延和最低时延下的最低成本扩容问题。其次, 采用贪婪取走算法, 以全局归一化时延增量作为取走候选链路标准, 从全拓扑网络的候选链路集中得到满足时延约束的新增链路。此外, 提出可以结合线性规划和贪婪取走算法实现时延约束下最低成本的新增链路及扩容。结果表明, 本文提出的方法可以有效解决针对时延优化的广域网络规划中拓扑设计和容量规划问题。

关键词: 网络规划; 网络容量确定; 贪婪取走; 新增链路; 时延优化

中图分类号: TN915

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2021137

Topology design and capacity planning of wide-area network based on delay constraint

XU Xiaoqing, TANG Hong, RUAN Ke, WU Juan, LIU Xiaojun

Research Institute of China Telecom Co., Ltd., Guangzhou 510630, China

Abstract: In order to optimize network performance and reduce construction cost, the topology design and dimensioning problems in wide-area network planning were investigated based on delay constraint with the use of linear programming (LP) and heuristic greedy-drop algorithm. Firstly, demands' weights were considered and LP was exploited with given paths to solve the capacity expansion problems of minimum delay and minimum delay with minimum cost. Secondly, greedy-drop algorithm was applied and candidate edges were removed from full mesh topology according to the global normalized delay increments. Thus, new added edges to satisfy the delay constraint were determined. Moreover, the combination of LP and greedy-drop algorithm was proposed to determine new added edges and capacity expansion of the delay-constraint scenario with minimum cost. The results turn out that the proposed algorithms are effective to solve the problems of topology design and capacity planning in wide-area network planning targeting delay optimization.

Key words: network planning, network dimensioning, greedy-drop, added edge, delay optimization

1 引言

如何让IP网络的拓扑设计和容量规划高效满足互联网业务高速增长需求,是基础电信运营商基础网络建设和运营部门面临的重大挑战,包括以下3方面。

- 架构设计变革后的评估和优化困难:经典的分层汇聚式网络架构,面临核心汇聚节点压力过大、非核心省份间时延性能差等问题,无法持续;而转向全面扁平化网络后,则缺少精确的量化评估手段以实现拓扑的最优化设计和最优化变更。
- 路由优化效率较低:运营商IP广域网的路由设计一般依赖于网络设备的分布式路由协议,无法满足面向业务/客户的、定制化的复杂约束条件,也难以保障网络利用率的均衡,导致IP骨干网络利用率一般只能在50%上下,网络建设成本高。
- 多维协同困难:IP广域网络的结构设计与成本约束与底层传输网络关系巨大,仅依赖IP网络层的约束无法实现整体性能和成本的最优。

具体到日常的网络容量规划中,存在多种需求场景,包括但不限于以下几种。

- 快速扩容场景:主要满足突发性业务需求,规模一般较小;实际操作时,约束因素相对较少,可快速满足业务需求。
- 常规扩容场景:运营商根据每年的业务发展预期,按需对现有网络资源进行增加或调整。
- 多约束扩容场景:在扩容过程中,会面临很多实际的约束,比如投资规模限制、是否允许优化拓扑、是否有特定业务时延要求等。

对运营商的网络规划,通常需要考虑时延、带宽、链路利用率和投资成本等方面。随着低时

延网络应用业务的涌现,例如游戏、直播、高清会议、电子交易等,时延成为网络规划任务中的一个重要因素。此外,成本是必须考虑的一个重要方面,需要在保障业务需求的同时,尽可能地降低网络建设成本。本文从时延/成本优化方面改进传统的IP骨干网拓扑设计与容量规划,推动实现广域IP网络设计过程的智能化与自动化,以期解决IP网络演进中面临的部分挑战。本文对容量规划问题进行数学建模,主要考虑传输时延的约束,分为拓扑固定和拓扑可变网络扩容两类问题,可以覆盖部分上述业务场景。

在一些网络规划文献中,时延的定义比较复杂,除了链路上的流量,还与链路所剩容量或者负载相关^[1-3]。在骨干网网络规划中,时延影响最大的是传输时延,占据光网络电路时延的90%以上^[4]。所以从运营商网络规划的角度,用路径长度来表征时延是一种比较合理和直观的方式,在本文中只考虑路径长度对应的传输时延。在降低网络时延的研究中,参考文献[5-6]提出通过升级节点或者链路方法,认为升级后的节点或链路会有一定比例的时延下降,从而降低整个网络的时延。但是升级前后各个节点时延下降比例难以比较合理地确定。参考文献[7]提出了在候选链路集中添加 k 个链路来降低所有需求对的平均最短路径,但这样只考虑了新增链路前后的所有需求的路径绝对长度的变化,没有考虑具体链路成本的不同。

通常,涉及目标函数为线性的优化问题,可以利用线性规划求解。传统上,大量的网络规划也可以建模成线性规划问题从而求得最优解。一般做法是提前对需求给定候选路径,每一个路径对应一个待分配流量。但涉及拓扑改变问题时,候选路径是动态变化的,不应该提前给定。而如果不提前对需求先给定候选路径,随着拓扑的增大,潜在的路径数迅速增加,给求解带来困难^[1]。另外,一旦涉及变量值只能涉及离散值,则线性



规划变成整数规划,随着问题规模的增大,求解难度迅速增加,大多是 NP-hard 问题,只能采用一些启发式算法来进行求解。本文针对运营商网络规划的需求和特点,从实际业务角度出发,定义了带权重的传播时延(流量×路径长度)和全局归一化时延(原始拓扑下各个需求的最短路径长度对应时延 1),针对拓扑不变和拓扑改变的情形,主要应用了线性规划和贪婪算法来建模和求解。

相比传统的做法,在应用线性规划求解最低时延时,本文不但考虑传输时延(路径长度),还加入需求量的影响,最优化需求量与路径长度的乘积。在应用贪婪算法寻找新增连接时,考虑了全局归一化时延的影响。首先研究了相对简单的拓扑固定情况下的最低时延和满足时延约束的最低成本的场景。在此基础上,考虑改变网络拓扑,新增网络连接降低整体时延;进一步研究满足时延约束下成本最低新增网络连接,即最低成本的新增局向问题。

2 拓扑不变的网络扩容

将网络表示为一个无向图 $G(V, E)$, V 表示网络节点, E 表示节点间的链路。采用路径建模(path formulation)的方法来研究网络规划问题,即对各个需求,提前确定候选路径,每个需求的各条路径对应的流为一个变量^[1]。下面研究目标函数为最低时延及最低时延下的最低成本情形,同时满足各个需求的带宽要求。其中,这里的“最低”是指给定候选路径下所能达到的最低。

2.1 求解最低时延

本文中只考虑路径长度带来的时延。同时,从网络运营的角度,更为合理的是考虑需求值的权重,对节点间最短路径进行加权,即对时延进行加权,用需求的带宽×需求点之间最短路径长度来表示。如图 1(a)所示,从 A 到 B 的需求是 1 Gbit/s,最短路径长度是 1 km,从 A 到 C 的需求是 99 Gbit/s,最短路径长度是 100 km。图 1(b)中, A 到 B 的

需求是 99 Gbit/s, A 到 C 的需求是 1 Gbit/s,路径长度和图 1(a)相同。虽然从 A 点出来的流量都是 100 Gbit/s,但从网络运营的角度,图 1(b)的总体时延要低于图 1(a)的,因为图 1(b)的大部分流量都用了比较短的路径,而为图 1(a)的大部分流量都用了比较长的路径。

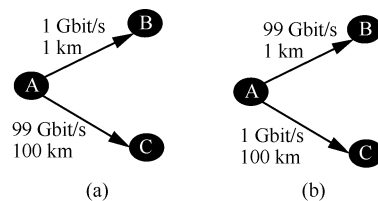


图 1 不同需求的时延比较

本节研究的最低时延是总体的最低时延,即让尽量多的流量通过比较短的路径。所以不但需求对应的节点间路径长度与时延有关,需求的值也有影响,所以当考虑最低时延时,最小化的是 $\sum \text{需求量} \times \text{路径长度}$,而不是最小化需求中的最大时延。

下面考虑目标函数为总体时延最低的网络扩容问题:假设有 D 个需求,对每个需求 d 先算出 P 条(例如 $P=4$)最短路径为候选路径,每条路径长度为 l_{dp} ,每条路径分配带宽为 x_{dp} ,下标 d 表示第 d 个需求, h_d 表示该需求的值, p 表示这个需求 P 条候选路径中的第 p 条路径。链路原有容量 c_e ,链路新增容量变量 y_e , e 表示第 e 条链路,共有 E 条链路。在网络规划中,为了考虑现网需求存在的突发性和波动性,往往要求链路的利用率低于一定水平,这里设置为 80%。最小化总体时延,该问题的线性规划建模如下:

$$\min F = \sum_{d=1}^D \sum_{p=1}^P x_{dp} \times l_{dp} \quad (1)$$

$$\text{s.t. } \sum_{p=1}^P x_{dp} = h_d, d = 1, 2, \dots, D \quad (2)$$

$$\sum_{d=1}^D \sum_{p=1}^P \delta_{edp} x_{dp} \leq 0.8(y_e + c_e), e = 1, 2, \dots, E$$

$$x_{dp} \geq 0, y_e \geq 0 \quad (3)$$

当 x_{dp} 对应的路径包含 e 链路时, $\delta_{edp}=1$, 其余时候为 0。

式(1)是目标函数,是加权的需求和路径长度的乘积,即对应加权的时延。式(2)表示每个需求 P 条路径的流量之和等于需求值,需求得到满足。式(3)表示对每段链路,其承载的各个需求的流不超过其新增的容量和固有容量之和的 80%。如果对某些链路容量有限制,则可以设置相应的链路容量变量 y_e 处在某些范围。

这样可以求解出固定拓扑情形下的满足需求并且时延最低的扩容问题的解,即各个需求的路径及带宽分配。其实,最低时延问题,本质上是对各个需求从候选路径表中挑选出一条最短路径(关于距离),然后让该需求要求的带宽都由该路径承担。这样,可以计算出该条路径上每段链路需要分担的带宽,把所有需求在该段链路的带宽相加,也就确定了该段链路的容量。这称为最短路径分配的方法^[1],可以不用求解上述的线性规划问题。但是,这种方法比较适用于链路容量从 0 开始设计扩容问题。真实的网络规划中,链路中已经有一些容量 c_e ,更适合用上述的线性规划求解。原因在于,最短路径分配的方法只用到一条路径,而上述线性规划中采用多条路径,还有链路利用率的约束,假设一个需求存在两条同样长度路径,采用上述线性规划,可能会用到两条路径,可以充分利用现有的链路容量,提高链路的利用率,不必新增太多容量,降低成本。如图 2 所示,所有链路的长度都是 1 km,固有容量 1 Gbit/s。从 A 到 D 有路径 A-B-D 和路径 A-C-D 相等。如果 A-D 的需求为 4 Gbit/s,如果候选路径取一个路径 A-B-D,则该路径需要扩容 3 Gbit/s,而该路径有两段链路总共需要扩容 $3 \times 2 = 6$ Gbit/s。而如果选取两个路径,每个路径只需扩容 1 Gbit/s,两个路径共有 4 段链路,需扩容 $1 \times 4 = 4$ Gbit/s,相比单路径情形减少了 2 Gbit/s 扩容,降低了成本。

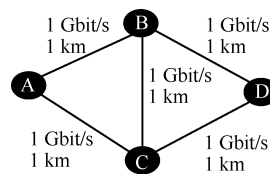


图2 一个需求可能存在多个相等路径

2.2 求解最低时延下的最低成本

由于本文采用了路径建模,路径中可能存在两条或多条路径时延相同,但成本不一样,因此可能在第 2.1 节中最低时延扩容问题的基础上进一步求出最低成本。同理,可以在最低成本的基础上,求解最低时延。在第 2.1 节中求解出最低时延,将其作为一个约束条件,将目标函数改成链路成本,就可以求解出最低时延下的最低成本, ρ_e 为链路 e 的成本系数。线性规划建模如下:

$$\min F = \sum_{e=1}^E \rho_e y_e \quad (4)$$

$$\text{s.t.} \sum_{d=1}^D \sum_{p=1}^P x_{dp} \times l_{dp} \leq \text{delay}_{\text{need}} \quad (5)$$

$$\sum_{p=1}^P x_{dp} = h_d, d = 1, 2, \dots, D \quad (6)$$

$$\sum_{d=1}^D \sum_{p=1}^P \delta_{edp} x_{dp} \leq 0.8(y_e + c_e), e = 1, 2, \dots, E$$

$$x_{dp} \geq 0, y_e \geq 0 \quad (7)$$

当 x_{dp} 对应的路径包含 e 链路时, $\delta_{edp}=1$, 其余时候为 0。

如果把式(5)最低时延约束修改成其他时延值约束,则该线性规划是更普遍形式的时延约束下的最低成本问题。当然,该线性规划求解结果可能和第 2.1 节单纯考虑最低时延情形下的成本一样,没有进一步降低。因为可能各个需求的最短路径就只有一条,而为了达到时延要求则必须选择这条路径。不过,在现网中,由于实际的网络比较复杂,对于一个需求往往存在多条路径要求,并且满足时延约束。因此很有可能在满足时延要求的条件下,增加候选路径,进一步降低扩



容成本。用一个例子进行说明,输入中国电信某骨干网 31 省份的部分需求和拓扑,链路成本系数 ρ_e 只简单地设置成与链路长度成正比,注意在真实规划中要根据链路实际情况设定。应用线性规划式(4)~式(7),求解不同候选路径数目下的最低时延下的最低成本,并进行归一化比较(以候选路径数为一条情况下的成本为 1),如图 3 所示。可以看出,在这个例子中,初期随着候选路径的增多,扩容成本降低,原因如上所述。当候选路径数超过 12 时,成本不再降低,原因在于路径长度要满足时延约束要求,否则即使有更多的候选路径,如果其不满足时延约束要求,在求解时延约束下的最低成本的过程中也不会选择这些路径,因此当路径数超过一定值时,成本趋近平稳,无法再下降。

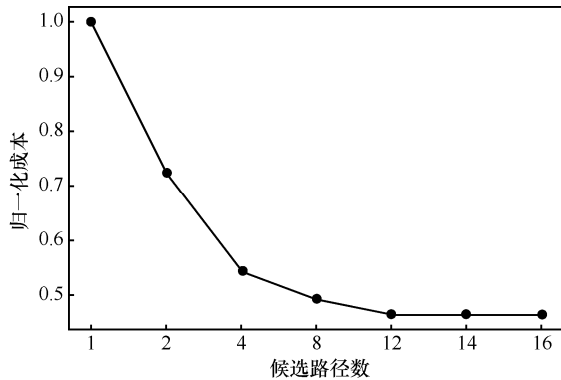


图 3 候选路径数对扩容成本的影响

3 拓扑改变的网络扩容

在前面的讨论中,网络拓扑是固定不变的。本文求解了扩容满足需求的情形下的最低时延问题,得出了最低时延下的各个路径的带宽分配。但是如果要求进一步降低时延,则需要在原来的拓扑上新增节点间的连接,以缩短某些需求的路径长度,从而降低总体的时延。

有约束的网络拓扑优化问题往往是 NP-hard 或者 NP-complete 问题,即随着问题规模的增大一般无法在多项式时间内求得精确

解,只能用启发式算法例如遗传算法、模拟退火等求得近似解^[1]。在时延约束的骨干网拓扑规划中,一般要求新增的链路数为最少。最小生成树算法能够求解最少连接问题,但是无法同时满足指定的时延要求。参考文献[7]提出了新增链路来降低整体平均最短路径的方法,但不是降低全局归一化时延。同时,其证明了该问题是 NP-hard 问题。其中,选择新增链路采用的贪婪方法是每次从候选链路集中添加选择链路,使得所有需求的平均路径长度下降最多。另外,离散选址的 p 中值问题中,研究从候选设备点集中挑选 p 个设备点,每个需求点连接到一个设备点,使得需求点与设备点距离或运输成本最小,也是 NP-hard 问题,其中贪婪取走算法被证明是一种有效求解方法^[8-9]。

类似地,本文采用全连接下的贪婪取走算法来确定新增链路。不同的是,这里以全局归一化时延增量为取走的衡量标准,从全连接的拓扑开始逐步删除对全局时延影响最小的新增候选链路。所谓全连接是指将所有节点互相连接;相比原始拓扑,多出来的链路为新增候选链路。另外与之前第一部分中的加权时延不同的是,对各个需求的时延分别进行了归一化:设每个需求在原始拓扑下的时延为 1。即原始拓扑下,某个需求 demand 的最短路径长度为 $\text{path_length}_{\text{demand},0}$,对应时延为 1。全连接拓扑下,取走第 $i-1$ 个链路后,新拓扑下该需求的最短路径长度 $\text{path_length}_{\text{demand},i-1}$,则新拓扑下该需求的时延 $\text{path_length}_{\text{demand},i-1} / \text{path_length}_{\text{demand},0}$ 。同样地,取走第 i 个链路该需求的时延为 $\text{path_length}_{\text{demand},i} / \text{path_length}_{\text{demand},0}$,则取走第 i 个链路后,相比上一轮(取走第 $i-1$ 个链路),所有需求的加权时延增量,即全局归一化时延增量为:

$$\Delta \text{delay}_i = \sum_{\text{demand}} \left(\frac{\text{path_length}_{\text{demand},i}}{\text{path_length}_{\text{demand},0}} - \frac{\text{path_length}_{\text{demand},i-1}}{\text{path_length}_{\text{demand},0}} \right) \quad (8)$$

相比直接用所有需求的最短路径长度的增量 $\Delta \text{delay}_i = \sum_{\text{demand}} (\text{path length}_i - \text{path length}_{i-1})$ 来作为取走链路的衡量标准, 归一化时延的好处如下。

- 对全局各个需求的时延进行了均衡处理: 在取走链路时, 不会只聚集于取走长度比较小的链路。例如边远省份节点, 一般与其连接的链路长度都比较大, 如果以绝对值来作为取走的标准, 它们将大概率被保留, 而一些距离比较小的节点间链路将更可能被取走。
- 能够方便地调节相关需求的权重。通过归一化时延, 每个需求的最大时延都为 1, 处于相同水平。在此基础上, 一些质量要求高的需求, 可以对其时延乘以一个大的权重, 使得该需求保持较低时延。

3.1 贪婪取走算法确定新增链路

根据上面所述, 采用归一化时延, 以全部需求归一化时延的增加程度为删除标准, 能够删掉对全局影响最小的链路, 只保留下重要的链路。假设归一化时延约束值为 $\text{delay}_{\text{set}}$, 需求数量为 N 。满足特定时延约束的贪婪取走链路算法的流程和具体步骤如下。

步骤 1 输入原始拓扑, 找出所有节点的全连接, 减去原始拓扑下的连接, 得出全连接下的新增候选链路列表 $[\text{link}_1, \text{link}_2, \dots, \text{link}_n]$ 。
 $\text{delay}_{\text{last_round}}$ 初始值为全连接拓扑下的全局归一化时延: 采用 Dijkstra 算法在原始拓扑下先找出每个需求的最短路径长度 $\text{path_length}_{\text{demand},0}$, 对应时延为 1; 在全连接拓扑下找出每个需求的最短路径长度 $\text{path_length}_{\text{demand},\text{last_round}}$, 则某需求的归一化时延为 $\text{path_length}_{\text{demand},\text{last_round}}/\text{path_length}_{\text{demand},0}$, 再汇总所有需求的归一化时延得到全局归一化时延 $\text{delay}_{\text{last_round}}$:

$$\text{delay}_{\text{last_round}} = \sum_{\text{demand}} \frac{\text{path length}_{\text{demand},\text{last_round}}}{\text{path length}_{\text{demand},0}} \quad (9)$$

步骤 2 从候选链路列表中逐步去掉一个新增链路 link_i (将候选链路列表该位置置空, $\text{link}_i = []$)。先判断其是否在删除列表或保留列表中。如果是, 则跳过此链路, 全局归一化时延增量列表对应位置 $\Delta \text{delay}[i]$ 设置为最大时延增量 (可设为需求的数量, 因为每个需求的时延最大设为 1), 即 $\Delta \text{delay}[i] = N$ 。如果不是, 去掉这个新增链路 link_i , 计算去掉后新拓扑下的全局归一化时延增量 Δdelay_i 和归一化时延 delay_i 。 Δdelay_i 和 delay_i 的计算同样采用 Dijkstra 算法先找出每个需求的最短路径 $\text{path_length}_{\text{demand},i}$, 分别得到各个需求的归一化时延 $\text{path_length}_{\text{demand},i}/\text{path_length}_{\text{demand},0}$, 再汇总所有需求的归一化时延得到全局归一化时延 delay_i :

$$\text{delay}_i = \sum_{\text{demand}} \frac{\text{path length}_{\text{demand},i}}{\text{path length}_{\text{demand},0}} \quad (10)$$

相比上一轮的增量为 $\Delta \text{delay}_i = \text{delay}_i - \text{delay}_{\text{last_round}}$ 。如果 $\text{delay}_i > \text{delay}_{\text{set}}$, 不满足时延约束, 则直接设置 Δdelay_i 为最大, $\Delta \text{delay}_i = N$ 。更新全局归一化时延增量列表对应位置 $\Delta \text{delay}[i] = \Delta \text{delay}_i$ 。再把这个已删除的新增链路 link_i 重新添加回候选链路列表。

步骤 3 重复步骤 2, $i=i+1$, 直至遍历所有 n 个候选新增链路, 得到此轮全局时延增量列表 $\Delta \text{delay} = [\Delta \text{delay}_1, \Delta \text{delay}_2, \dots, \Delta \text{delay}_n]$ 。

步骤 4 判断是否满足 $\text{delay} \leq \text{delay}_{\text{set}}$, 即是否 $[\Delta \text{delay}_1, \Delta \text{delay}_2, \dots, \Delta \text{delay}_n]$ 中每个元素都等于 N , 若是则算法结束, 候选链路列表中剩下的新增链路即为所求, 否则进行下一步。

步骤 5 先将 $[\Delta \text{delay}_1, \Delta \text{delay}_2, \dots, \Delta \text{delay}_n]$ 中为 0 的 Δdelay_i 对应的 link_i 去掉 (这些链路对全局时延没有影响, 也可以设置一定的阈值, Δdelay_i 低于此阈值对应的链路都去掉), 添加进删除列表, 并将为 0 的 Δdelay_i 重置为 $\Delta \text{delay}_i = N$ 。再从剩余的全局时延增量列表中, 找到 Δdelay 最小值对应的链路, 即 $m = \arg \min([\Delta \text{delay}_1, \Delta \text{delay}_2, \dots,$



Δdelay_n])。如果有多个最小值,则取这些最小值中对应链路长度最长的 m 。去掉 link_m 对全局时延增加影响最小,所以确定删掉 link_m , 将其添加进删除列表。更新去掉 link_m 后的全局归一化时延 $\text{delay}_{\text{last_round}} = \text{delay}_m$ 。

步骤 6 重复步骤 2~步骤 5,直至为满足 $\text{delay} \leq \text{delay}_{\text{set}}$ 不能再继续删掉新增连接,算法结束,候选链路列表中剩下的新增链路即所求。

3.2 实验结果及分析

为证明贪婪取走算法的有效性,采用另外一个公开的网络拓扑 GEANT (2001 年)^[10]进行验证。在 GEANT 网络中,有 27 个节点,38 段链路。由于只有节点的经纬度信息,所以直接用节点间的球面距离近似作为节点间链路的长度。假设 27 节点间都存在连接需求,则有 $27 \times 26 = 702$ 个需求,全局归一化时延最大为 702。将所有节点都连接,在全连接拓扑下计算时延(每个需求时延=此时最短路径/原来最短路径),得到全局归一化时延最小为 516。所以全局归一化时延最多能下降至 $516/702 = 73.5\%$ 。全连接下的链路总数 351,减去现有 38 条链路,候选新增链路数为 313 条。设置归一化时延约束为原始拓扑下全局归一化时延的 95%、90%、85%、80%、75%,即 $702 \times 0.95 = 666.9$, $702 \times 0.9 = 631.8$, $702 \times 0.85 = 596.7$, $702 \times 0.8 = 561.6$, $702 \times 0.75 = 526.5$,将贪婪取走算法与模拟退火算法,以及逐步取走候选新增连接中最长连接的贪婪算法(这里称为直接贪婪算法)得到的结果进行对比。

直接贪婪算法中,在满足时延约束要求的情况下,每次取走候选新增连接中长度最长的连接,所以最后会剩下那些相对较短的连接来满足时延约束。

模拟退火的初始温度设置为 100,温度衰减系数为 0.98,终止温度为 0.001。每个温度下产生 200 次新解。新解为随机产生一个新增链路,或者删除一个新增链路。判断新解优劣的标准是新增

链路数是否小于旧解并满足时延约束。满足归一化时延约束时,新增链路数小于旧解,则新解优于旧解,接受新解,否则以一定概率接受新解。如果新解不满足归一化的时延约束,则新解被接受的概率为 0。初始解设为所有新增链路,即初始解是全连接。

不同归一化时延约束下,3 种算法得到的新增链路数如图 4 (a) 所示。基本上贪婪取走算法得到新增链路数略少于模拟退火得到的解,部分情况下得到相等数量的新增链路,不过具体的链路不一样,此外贪婪取走算法运行速度比模拟退火算法快很多。而要降到相同的时延,相比模拟退火和贪婪取走算法,直接贪婪算法要增加更多的新增链路。

再对比这些新增链路的总长度,因为新增链路数一样并且满足时延约束情况下,从网络规划角度,新增链路总长度越小越好。在相同时延约束比例,基本上贪婪取走算法得到的新增链路总长度比模拟退火和直接贪婪算法得到的要小,如图 4 (b) 所示。

基于另一个中国电信骨干网的部分拓扑数据,时延约束设置为全部需求的归一化时延小于原始拓扑下的 97%、95%、93%、90%、88%,求满足这些时延约束的最少新增连接。这个骨干网有 31 个逻辑节点,原始基础拓扑约有 75 条逻辑链路。将这些节点全连接,大约有 400 条新增连接。采用同样的方法,得到基本类似的结果,如图 4 (c) 和图 4 (d) 所示,再次证明贪婪取走算法确定新增连接的有效性和可行性。

相比直接贪婪算法,贪婪取走算法在优化效果上取得了较好效果(较少的新增链路数和新增链路长度);相比模拟退火,优化效果接近,但贪婪取走算法的计算速度较快。综合以上两个例子,证明贪婪取走算法是一种比较有效的指定时延约束下的优化网络拓扑方法。

网络规划中往往还有可靠性的要求,例如规定对每个需求,除了最短路径,与其边不相交的

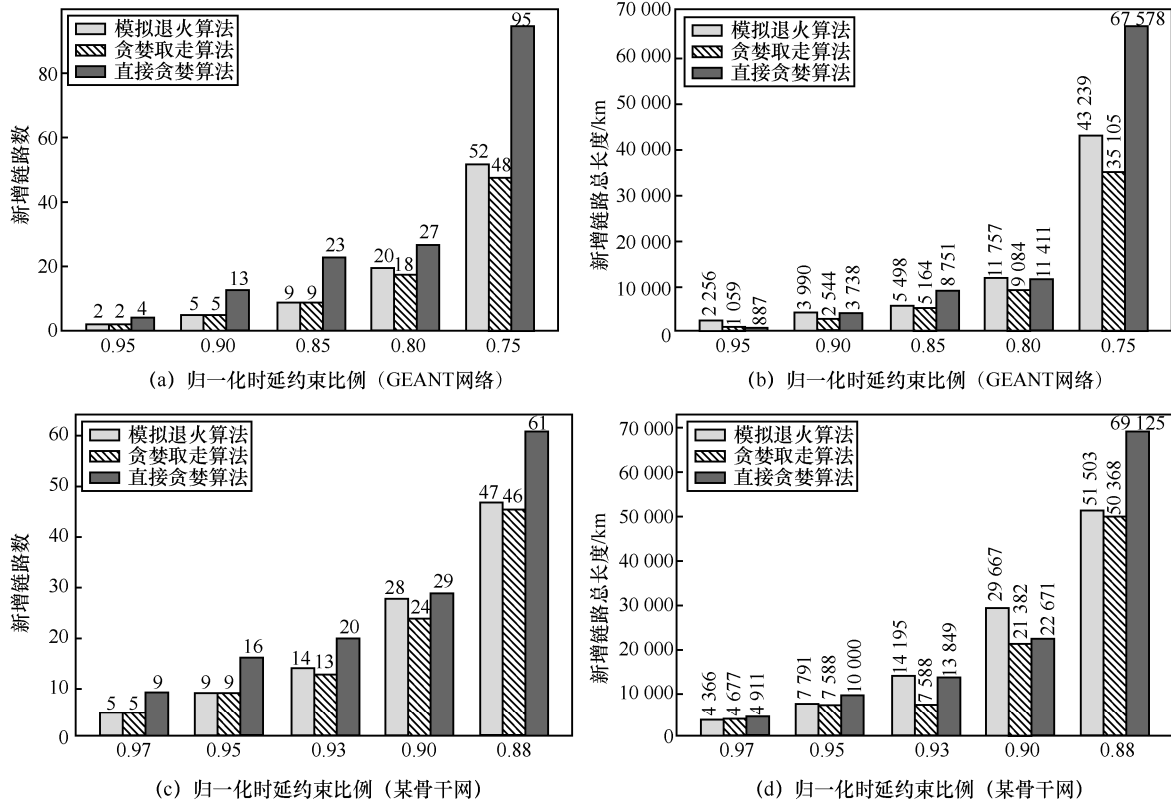


图4 3种算法得到的新增链路数和链路总长度对比

第二最短路径也要保证处于一定的时延约束,进行第二路径的时延约束。此外,还可以分别对各个省份节点的所有需求进行约束,使其低于一个界限,进行省份时延约束。可以按照实际需求,设置更多层次的QoS约束条件,然后在算法中步骤2中进行添加,其他不变。这样在删除候选链路时必须保证QoS约束成立,否则不删除,最终得到满足多QoS约束下的新增连接。不过,随着QoS约束条件的增加,在决定是否删除候选链路时需要进行多次判断,运算时间也会增加。

3.3 贪婪取走算法+线性规划的结合

对于一般的、只要求归一化时延约束下的最少连接问题,贪婪取走算法可以有效解决,并且得到节点之间的新增连接。但是,由于只考虑了使得新增链路数量最少,不涉及具体链路成本,同时考虑的是归一化时延,比较适合用于网络拓扑的优化设计。在本文规划中,主要用于逻辑节

点层面的新增连接。如果要进一步研究扩容问题,则要更精细地考虑物理节点层面的成本问题,需考虑各个需求值的大小,各段链路的固有容量及每段链路的成本系数。同时,不考虑归一化时延,而是考虑加权时延约束。然后结合第一节中的线性规划建模方法,可以求解得到满足加权时延约束下的最低成本的新增连接。即贪婪取走算法先得到一个候选链路集(逻辑节点层面),再将逻辑节点层面的候选链路集对应成物理节点层面的链路集,再继续用贪婪取走的思路,结合线性规划从候选链路集中继续挑选新增链路(物理节点层面),使得新增链路后,即满足加权时延约束又达到扩容成本最低。

这里分成两步得出最终的新增链路,而不是用贪婪取走算法结合线性规划一步得出,是因为在贪婪取走算法中每轮求解多个线性规划也相对耗时,所以先不考虑线性规划,可快速地得出比



较少数量的候选链路集,在此基础上再用贪婪取走算法+线性规划,每轮可以求解比较少的线性规划问题,等于候选链路数目。

在线性规划式(1)~式(3)中,得到最小加权时延 $\text{delay}_{\min}$ 后,想进一步下降 $\text{delay}_{\min}$, 设下降比例为 ratio , 即 $\text{delay}_{\text{need}} = \text{delay}_{\min} \times \text{ratio}$ 。注意 $\text{delay}_{\min}$ 和 $\text{delay}_{\text{need}}$ 是加权的时延,不是归一化的时延。可以先设定归一化时延的约束 $\text{delay}_{\text{set}}$, 然后用贪婪取走算法先计算得出候选链路列表,再运行最低时延的线性规划算法式(1)~式(3),判断在候选链路列表下是否能满足加权的时延要求 $\text{delay} \leq \text{delay}_{\text{need}}$ 。如果不能,说明归一化时延的约束值设置得较大,需重新将约束值调小,才能产生更多的候选链路,从而更有可能使得加权时延降低以满足约束。如果 $\text{delay} \ll \text{delay}_{\text{need}}$ 则说明候选链路过多,说明归一化时延的约束值设置得较小,需重新将约束值调大,避免产生太多候选链路。当 delay 小于并接近 $\text{delay}_{\text{need}}$, 继续考虑贪婪取走算法结合线性规划算法。每轮去掉一个候选链路,基于当前拓扑用线性规划式(4)~式(7)计算最低成本(目标函数要多加一项新增链路开通成本),遍历后删除最小成本(成本下降量最大)对应的取走链路,直至为满足加权时延约束,不能再删除任何一个候选链路为止。

仅使用贪婪取走算法属于“粗调”,贪婪取走算法加上线性规划算法属于“细调”。当考虑加权时延约束,并且需要考虑需求流量大小和链路成本等时,贪婪取走算法结合线性规划算法,可以更为合理地解决新增链路问题,最终得到新增链路下各个需求在各个候选路径的带宽分配以及链路需要增加的容量。另外,也可以在贪婪取走算法得到的候选连接下,直接进一步考虑线性规划算法式(4)~式(7),求出成本最低的满足时延约束的扩容,不再逐轮考虑删除候选连接,但这样可能多开通了一些新增连接,成本会比较高。

4 模型效果评估及展望

评估网络方案的优劣是一个复杂的工作^[10],本文主要从资源优化角度考虑此问题,运营层面的评估不在本文的考虑范围内,从以下角度来评估资源优化方案。

- 资源分配的效率,也就是说能用最少的资源满足业务需求和各类业务层面的约束条件。资源分配效率的评估指标相对简单,主要关注总的资源使用量,由于采用了线性规划模型,那么模型的有效性地关键是定义成本函数;既要客观地评估实际的网络建设成本,同时还应考虑算法的可收敛性。
- 资源分配方案对网络性能的影响,针对某一个特定的网络性能指标,不同的资源分配方案对该性能指标的影响;本文主要关注时延数据,评估标准是时延性能的改善程度以及与物理最优时延的逼近效果。
- 资源分配的公平性^[12-13],也就是说,当出现资源抢占时,能以一种“公平”的方式合理地分配资源。当然,“公平”的定义方式可以是多样化、可定制的。

本文主要考虑前两点,并应用到实际的IP骨干网规划中。未来计划在以下几个方面继续增强算法能力和模型的全面性,以适应更复杂的业务场景。

- 路由优化场景,将容量规划与路由优化的结合,补充路由优化场景,解决短期内现网资源无法提供业务的场景。
- 量化的可靠性模型,现有的资源预留机制采用随机故障模型,并不能反映底层物理网络实际的故障概率。未来计划使用现网故障统计数据建立骨干网故障的概率模型,更加准确地判断故障特征,并提供资源预留。
- 公平性约束问题,当资源受限时,需要根据

一定公平性原则进行分配或扩容。计划研究公平性资源分配/扩容算法在IP骨干网的应用,探索适应互联网特点的层次化QoS模型,更加公平地为互联网客户分配资源。

5 结束语

本文详细地研究了网络规划的两个关联场景:给定固定拓扑情况下,求解满足时延约束的最低成本扩容;拓扑改变情况下,求解满足时延约束的新增链路拓扑设计和扩容。分别作出相关数学描述和提出解决算法,并结合示例以及现网数据进行分析。本文提出的方法可以应用到实际的网络规划和优化中,对运营商网络规划和优化有一定借鉴和启发意义。

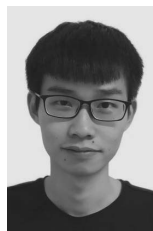
参考文献:

- [1] PIÓRO M, MEDHI D. Overview[M]//Routing, Flow, and Capacity Design in Communication and Computer Networks. Amsterdam: Elsevier, 2004: 3-36.
- [2] VUTUKURY S, GARCIA-LUNA-ACEVES J J. A simple approximation to minimum-delay routing[R]. Defense Technical Information Center, 1999.
- [3] GILGUR A, ECK B. Sources of Traffic Demand Variability and Use of Monte Carlo for Network Capacity Planning[J]. 2014.
- [4] 中国电信集团有限公司. 低时延光网络技术白皮书[R]. 2016.
China Telecommunications Group Co., Ltd. White paper of low latency optical network technology[R]. 2016.
- [5] MEDYA S, BOGDANOV P, SINGH A. Towards scalable network delay minimization[C]//Proceedings of 2016 IEEE 16th International Conference on Data Mining (ICDM). Piscataway: IEEE Press, 2016: 1083-1088.
- [6] MEDYA S, RANU S, VACHERY J, et al. Noticeable network delay minimization via node upgrades[J]. Proceedings of the VLDB Endowment, 2018, 11(9): 988-1001.
- [7] PAROTSIDIS N, PITOURA E, TSAPARAS P. Selecting shortcuts for a smaller world[C]//Proceedings of the 2015 SIAM International Conference on Data Mining. Philadelphia, PA:

Society for Industrial and Applied Mathematics, 2015.

- [8] 关怀庆, 张毕西, 欧江艳. 贪婪取走启发式算法在离散网络选址中的研究[J]. 系统科学学报, 2010, 18(3): 49-52, 56.
GUAN H Q, ZHANG B X, OU J Y. The study of greedy dropping heuristic algorithm in discrete network location[J]. Chinese Journal of Systems Science, 2010, 18(3): 49-52, 56.
- [9] REESE J. Solution methods for the p-Median problem: an annotated bibliography[J]. Networks, 2006, 48(3): 125-142.
- [10] 曹晓宏, 徐思雅, 周桂平, 等. 面向多层次、多指标的光网络规划与评估方法[J]. 电信科学, 2019, 35(9): 58-68.
CAO X H, XU S Y, ZHOU G P, et al. Multi-level and multi-index planning and evaluation method for optical network[J]. Telecommunications Science, 2019, 35(9): 58-68.
- [11] NACE D, PIORO M. Max-Min fairness and its applications to routing and load-balancing in communication networks: a tutorial[J]. IEEE Communications Surveys & Tutorials, 2008, 10(4): 5-17.
- [12] JAIN S, KUMAR A, MANDAL S, et al. b4[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 3-14.

[作者简介]



徐晓青(1990—),男,博士,中国电信股份有限公司研究院工程师,主要研究方向为互联网及电信网络技术、人工智能和大数据等。

唐宏(1974—),男,中国电信股份有限公司研究院,教授级高工,中国电信IP首席专家,主要研究方向为互联网及电信网络技术、网络新技术、人工智能等。

阮科(1979—),男,中国电信股份有限公司研究院工程师,主要研究方向为IP网络架构设计及最优化规划方法。

武娟(1977—),女,中国电信股份有限公司研究院高级工程师、主任,主要研究方向为云计算、互联网及电信网络技术、大数据、人工智能等。

刘晓军(1977—),男,中国电信股份有限公司研究院工程师,主要研究方向为云计算、互动媒体技术、核心网络大数据分析与人机智能等。