引用本文：王为亮，谭绍锋，肖雁鹏. 一种基于光网络的搜索 K 最短路径的 Yen 改进算法[J]. 光通信技术，2022，46(4)：101-106.

一种基于光网络的搜索 *K* 最短路径的 Yen 改进算法

王为亮，谭绍锋，肖雁鹏

（中国电子科技集团公司 第三十四研究所，广西 桂林 541004）

摘要：为提高 *K* 最短路径(KSP)算法中路径计算的效率和规划路径的相异性，首先介绍了光网络的图论描述、路径相近性定义和平行边的理论，然后对 KSP 问题和传统 Yen 算法进行了简单描述，分析了 KSP 算法研究现状，最后提出一种 Yen改进算法，重点阐述了相异路径计算策略和 Yen 改进算法实现步骤。通过构建与实际生产环境类似的拓扑图，对 Yen 改进算法进行验证，并与其它算法进行路径相近性和计算时间对比，证明了其有效性。

关键词：*K*- 最短路径；Yen 算法；光网络；相异路径

中图分类号：TN256　**文献标志码**：A　**文章编号**：1002-5561(2022)04-0101-06

DOI：10.13921/j.cnki.issn1002-5561.2022.04.019　**开放科学(资源服务)标识码(OSID)**：

Yen improved algorithm for searching *K*-short path based on optical network

WANG Weiliang, TAN Shaofeng, XIAO Yanpeng

(The 34th Research Institute of CETC, Guilin Guangxi 541004, China)

Abstract: In order to improve the efficiency of path calculation and the dissimilar of planning paths of *K*-short path (KSP) algorithm. First, this paper introduces the graph theory description of optical network, definition of path similarity and parallel edges theory. Then it describes the KSP problem and the traditional Yen algorithm briefly, and analyzes the research status of KSP algorithm. Finally, the paper proposes an Yen improved algorithm, and expounds the calculation strategy of dissimilar paths and the implementation steps of the improved algorithm emphatically. By constructing a topology map similar to the actual production environment, the Yen improved algorithm is verified, and the path proximity and calculation speed are compared with other algorithms to prove its effectiveness.

Key words: *K*-short path, Yen algorithm, optical network, dissimilar paths

0 引言

使用光网络传输用户业务时，为保证业务可靠传输，通常会为业务规划一条工作路径和多条保护路径。由工作路径切换到保护路径的时间通常为毫秒级，用户业务不会中断。如果所有传输路径(工作路径、保护路径)同时中断，用户业务会出现中断。因此，减小工作路径和保护路径同时中断的概率尤为重要。

在铺设光缆时，受地理环境和成本的限制，2 个通信节点之间的多根光纤可能放置在同一条光缆中。传统光路径规划算法通常未考虑路径在地理空间上的相异性，导致同一业务的多条光传输路径可能经过同一光缆，当光缆断开，用户业务中断的概率大大增加。所以规划光路径时，不能只考虑路径长短，还需考虑光传输路径在地理空间上的相异性。此外，当极端情况发生时，光路径计算速度应足够快，尽量缩短业务中断时间。

Dijkstra 算法通常被用于计算 2 个节点间的最短路径，在需要计算 *K* 条渐次最短路径的场景下，通常使用 *K* 最短路径(KSP)算法，但 KSP 算法搜索的 *K* 条最短路径往往存在路径相近性大的问题[1]。为此，本文对传统 Yen 算法进行改进，提高所规划的路径间的相异性，并采用启发式搜索[2]的方法，提高路径计算的速度。

收稿日期：2021-12-27。

作者简介：王为亮(1985—)，男，汉族，湖北咸宁人，硕士，工程师，从事光通信产品及应用软件的需求分析、设计和研发工作。主要研究方向为光通信网络、路由交换、数据库和软件定义网络等，曾负责或参与多项国家重点装备和项目的应用软件设计和研发工作。

1 对象模型

1.1 光网络图论描述

光网络中，设备可抽象为顶点，设备之间的物理链路抽象为边。则可将光网络拓扑图记为 $G=(V(G),E(G))$。其中，$V(G)$为图中所有顶点的集合，$E(G)$为图中所有边的集合，每一条边均有方向，即 G 为有向图。图 G 中的路径用 P 表示，P 中所有节点都不同时，P 被称为无环路径，也称为简单路径。

1.2 路径相近性定义

对于路径 P_1，记其长度为 $l(P_1)$。如有另外一条路径 P_j，令 P_1 与 P_j 的共同路段长度为 $l^a(P_j,P_1)$，不同路段长度为 $l^b(P_j,P_1)$，则有

$$l(P_j)=l^a(P_j,P_1)+l^b(P_j,P_1) \tag{1}$$

将 P_1 和 P_j 的相近性定义为

$$l_{similarity}=l^a(P_j,P_1)/l(P_1) \tag{2}$$

1.3 平行边

目前，研究 KSP 算法时使用的拓扑图都没有考虑平行边。文献[3]中提到平行边的定义，如图 1 所示。图中 e_5 和 e_6 即为平行边。工程建设时，对于多根光纤放置在同一条光缆中的情况，抽象成拓扑图就是 2 个节点间存在平行边。

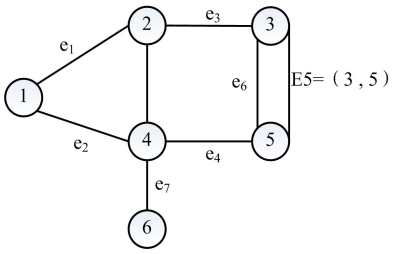

图 1 平行边示意图

2 KSP 问题和 Yen 算法简介

2.1 KSP 问题简介

KSP 问题是对最短路径问题的扩展，求解 KSP 问题[4]的目的是从一个图中搜索源节点和目的节点之间的多条最短路径，满足具体应用场景中对不同路径的选择需求。该问题可细分为两类：限定无环 KSP 问题和一般 KSP 问题。前者要求所求的路径不能含有环，后者对所求路径没有任何限制[5-6]。为应用于光通信网络领域，本文主要关注限定无环 KSP 问题的研究情况。

用 P_i 表示从源节点 s 到目的节点 d 的第 i 短路径，KSP 问题是要确定路径集合 $P_K=\{p_1,p_2,p_3,\cdots,p_k\}$，使其满足以下 3 个条件：$K$ 条路径是按次序产生的；K 条路径是按长度从小到大排列的；这 K 条路径是最短的。

2.2 传统 Yen 算法简介

传统 Yen 算法[7]是典型的限定无环 KSP 算法，采用偏离路径算法思想，即先计算出第一条最短路径，然后在此基础上依次算出其它的 $K-1$ 条最短路径。该算法搜索路径的基本步骤如下：

①输入起点 s、终点 d 以及需要搜索路径的数量 K。

②使用 Dijkstra 算法计算一条从 s 到 d 的最短路径，记为 P_1。

③搜索次最短路径。将 P_1 上除尾节点以外的所有节点确定为分离点，分离点记为 V_i，$i=1,2,3,\cdots,x$。

④遍历步骤③中确定的分离点，求 V_i 到 t 的最短路径，所求的最短路径放入集合 B 中。

⑤如果集合 B 不为空，从集合 B 中选择权值最小的路径放入集合 A 中，此路径记为 P_2，并将该路径从集合 B 中移除，P_2 为次最短路径。

⑥如果求解的路径数量小于 K 并且集合 B 不为空，跳转到步骤③，将步骤⑤中放入集合 A 的路径替换为 P_1，跳转到步骤③继续求解。若有更多轮的求解，依次类推。

⑦如果达到求解路径数量要求，或者集合 B 为空，则算法结束。

2.3 KSP 算法研究现状

目前，限定无环 KSP 问题的算法的研究主要集中在 2 个方向，一是偏离路径算法，二是改进 Dijkstra 算法。偏离路径算法方面，LAWLER E L 等人[8]对 Yen 算法进行改进，提升了算法效率；MARTINS E Q V 等人[9]对 Yen 算法进行了深入研究，提出了矩阵乘积态(MPS)算法和关于 Yen 算法的一种新的实现方法。HERSHBERGER J 等人[10]提出偏离路径算法，在理想情况下，算法计算速度较 Yen 算法有较大提升。改进 Dijkstra 算法方面，柴登峰等人[11]基于划分子图，递归调用 Dijkstra 算法的思路，提出了新算法；袁红涛等人[12]通过优化 Dijkstra 算法的过程，提升了算法效率；高松等人[13]改进 Dijkstra 算法，采用双向搜索的思路，提升了算法效率。综上所述，当前的 KSP 算法研究问题主要在于如何提高算法的计算速度和降低算法复杂度。

3 Yen 改进算法实现过程

不同于传统 Yen 算法的盲目搜索，本文提出的 Yen 改进算法使用启发式搜索方法，根据知情信息对拓扑图进行处理，减少节点数目。知情信息包括：已使用的路径、不可用的链路和节点，分别用数据结构 Set1、Set2 和 Set3 表示。进行路径计算前，先对拓扑图进行处理，可减少拓扑图中节点和边的数量，从而减少计算量，提高路径搜索效率。

3.1 相异路径计算策略

相异路径选择算法主要包括迭代惩罚法、通道最短路径法、极小极大法和离散p-扩散法等[14]。其中，通道最短路径法可能计算出环路，但计算稳定性不够，不适应通信领域；极小极大法计算相异路径效果不佳；相比极小极大法，离散 p-扩散法的优势主要体现在求解大量路径的场景，而本文不涉及求解大量路径，所以不考虑该方法。由于相异路径的计算方法是在迭代惩罚法的基础上改进得到的，且迭代惩罚法实现简单，在工程实践中有足够多的知情信息可供使用，还能规避算法搜索盲目性的缺点，因此本文选择迭代惩罚法评价路径的相异性，引入路径相异性计算公式，使路径相异性评价机制保持严密。考虑到算法实际运行的硬件环境(CPU 计算能力不强，但是内存空间充足)，本文采用空间换时间的思路，以增大内存空间为代价，将知情信息进行缓存，这不仅能提升计算路径的差异性，还能降低计算代价。

基于传统 Yen 算法，本文增加了单独计算完全节点偏离路径和完全路径偏离路径的子流程，以克服传统 Yen 算法计算出的路径集相近性大的问题。计算完全节点偏离路径时，需要将计算出的路径中出现的除首尾节点以外的所有节点从拓扑图中删除。计算完全路径偏离路径时，需要将已算出的路径中出现的边从拓扑图中删除。在不能计算出完全节点偏离路径和完全路径偏离路径时，使用传统 Yen 算法进行路径搜索，以保证路径计算不会出现遗漏。传统 Yen 算法进行分离点遍历时，仍然可以根据知情信息对拓扑图进行处理，减少分离点的遍历，提高搜索效率。

Yen 改进算法在使用传统 Yen 算法搜索路径时，尽量计算出足够数量的相异路径。从集合 B 选择路径时，基于式(2)，尽量选择与集合 A 中相近性小的路径，路径计算公式为

$$l_a=\min_{j\in B}\left[\sum_{i\in A}\frac{l^a(P_j,P_i)}{l(P_i)}\right] \tag{3}$$

分别计算集合 B 中的每条路径与集合 A 中每条路径的相近性并求和，和值记为 sim，sim 作为集合 B 路径的一个属性，然后选取集合 B 中 sim 值最小的路径放入集合 A。

3.2 Yen 改进算法描述

Yen 改进算法步骤如下：

①生成原始的图 G，确定求解路径的源节点 s、目的节点 d、知情信息(包括 Set1、Set2 和 Set3)和需要计算的路径数量 M。

②尝试计算第一条路径，也就是最短路径。计算时如果 Set1 不为空，则进入步骤③，否则进入步骤④。如果不需要计算，则跳转到步骤⑤。

③计算节点完全偏离路径，根据 Set1 对原始图 G 进行处理，将已用路径上的顶点全部删除，然后使用 Dijkstra 算法求解最短路径。若求解路径成功，路径存入集合 A，返回计算结果，本轮计算结束，等待下一轮计算；若求解路径失败，根据 Set1 对原始图 G 进行处理，将 Set1 的路径中包含的边删除，然后尝试计算链路完全偏离路径。如果计算成功，则将得到的路径存入集合 A；如果计算失败，则进入步骤④。

④不对原始图 G 进行处理，直接尝试使用 Dijkstra 算法求解最短路径。若求解路径成功，路径先放入集合 A，如果 Set1 不为空，则路径移入到集合 B；若求解路径失败，则说明此时不存在符合条件的从源节点到目的节点的路径，计算结束。

⑤尝试计算完全偏离路径。根据 Set1、Set2 和 Set3 中的信息，对原始图 G 进行处理，删除需要过滤的边、节点和路径。使用 Dijkstra 算法求解最短路径，如果计算失败，则跳转到步骤⑥；如果计算成功，将求解的路径放入集合 A 并返回，本轮计算结束，等待下一轮计算。

⑥计算偏离路径。此步骤和传统 Yen 算法步骤大概相似，不同之处在于，本文会在遍历偏离点列表时，先删除不可用的偏离点，提高计算效率。另外，从集合 B 中选取路径时，会考虑选取的路径和集合 A 中已有路径的相近性，计算方法参考式(3)。

算法详细的流程如图 2 所示。

4 算法仿真与分析

4.1 路径相近性比较

为比较传统 Yen 算法和 Yen 改进算法计算路径的相近性，本文构建的拓扑图如图 3 所示。其中 L 代表边的 id，W 代表边的权值，计算从源节点 s(图中①)到目的节点 d(图中⑧、⑨、⑩)之间的 3 条最短路径，计算结果如表 1 所示。其中，如果同一源节点到目的节点被计算出了多条相同的路径，则使用 L 进行区分，否则路径后不标出 L。

从表 1 可以看出，传统 Yen 算法在拓扑图中有平行边的情况下，计算出的最短路径相近性很大，极端情况下，计算出的前 3 条最短路径经过的节点完全一样。与之相比，Yen 改进算法计算出的前 3 条最短路径相近性小很多，路径中节点完全一样的情况很少。

本文再对极小极大法和 Yen 改进算法进行比较。

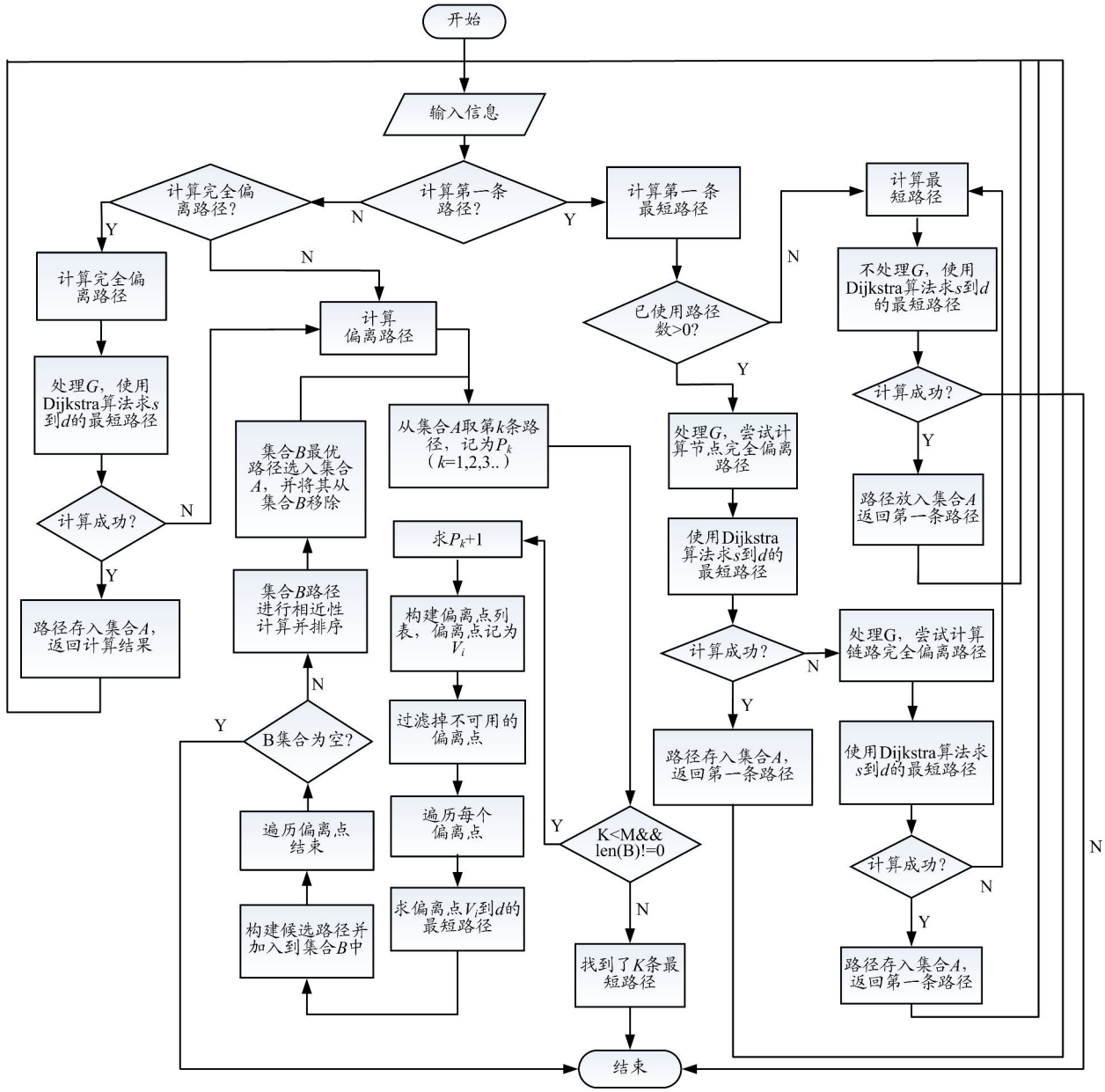

图 2　Yen 改进算法流程图

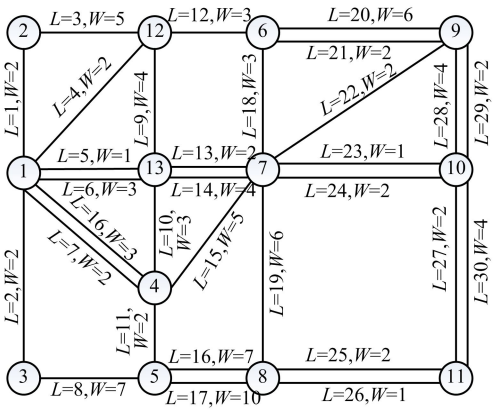

图 3　相近性比较所用拓扑图

表 1　2 种算法路径计算结果

$s \rightarrow d$	传统 Yen 算法计算路径（L）	Yen 改进算法计算路径（L）
①→⑩	①→⑬→⑦→⑩（5 13 23） ①→⑬→⑦→⑩（5 13 24） ①→⑬→⑦→⑩（6 13 23）	①→⑬→⑦→⑩（5 13 23） ①→⑫→⑥→⑨→⑩ ①→④→⑤→⑧→⑪→⑩
①→⑨	①→⑬→⑦→⑨（5 13 22） ①→⑬→⑦→⑩→⑨ ①→⑬→⑦→⑨（6 13 22）	①→⑬→⑦→⑨ ①→⑫→⑥→⑨ ①→④→⑤→⑧→⑪→⑩→⑨
①→⑧	①→⑬→⑦→⑩→⑪→⑧（5 13 23 27 26） ①→⑬→⑦→⑩→⑪→⑧（5 13 24 27 26） ①→⑬→⑦→⑧	①→⑬→⑦→⑩→⑪→⑧（5 13 23 27 26） ①→④→⑤→⑧ ①→⑫→⑥→⑦→⑧

使用 KSP 算法生成一个大的路径集合 C，然后依据可取性指数产生相异路径子集 DS。可取性指数定义中，β 值越大，越强调共同路段的极小化。如果 DS 的大小为 K，K 为求解的路径（设此时 K=5），那么 C 的大小为 $K+\delta$，其中 δ 是调节集合 C 大小的一个因子。极小极大法中，为了更详细考察算法特性，以探究其是否拥有比 Yen 改进算法更突出的优势，δ 取不同值；Yen 改进算法中，K 取值为 5 已完全满足应用需求，所以 Yen 改进算法只考虑 δ=0。采用极小极大法和 Yen 改进算法计算得到的 C 中相同路径数和总数的比例分别如表 2 和表 3 所示。

表 2 极小极大法计算结果

δ	C 中相同路径数和总数的比例											
	①→⑧				①→⑨				①→⑩			
	β=1	β=5	β=10	β=50	β=1	β=5	β=10	β=50	β=1	β=5	β=10	β=50
0	0.8	0.8	0.8	0.8	0.6	0.6	0.6	0.6	1.0	1.0	1.0	1.0
5	0.8	0.8	0.8	0.8	0.4	0.4	0.4	0.4	0.4	0.6	0.6	0.6
10	0.6	0.6	0.6	0.6	0.4	0.4	0.4	0.4	0.6	0.8	0.8	0.8
30	0.4	0.4	0.4	0.4	0.4	0.8	0.8	0.8	0	0.8	0.8	0.8
50	0.6	0.4	0.8	0.4	0.4	0.8	0.8	0.8	0	0.8	0.8	0.8
100	0.4	0.6	0.6	0.6	0	0.8	0.8	0.8	0	0.8	0.8	0.8

表 3 Yen 改进算法计算结果

δ	C 中相同路径数和总数的比例		
	①→⑧	①→⑨	①→⑩
0	0.6	0.4	0.4

对比表 2 和表 3 可以看出，极小极大法中，当 δ 递增、β 不变时，集合 C 中相同路径比例并不会呈减小趋势；当 δ 不变、β 递增，集合 C 中相同路径比例也并不会呈减小的趋势。这是因为图 G 中存在多条平行边时，集合 C 中会存在大量重复路径，如果这些重复路径的可取性指数是极大的，当计算第 2 条以后的路径时，极小极大法中选取的可取性指数对应的路径重复概率会增大。Yen 改进算法中，集合 C 中相同路径数占总路径数的比例整体小于极小极大法，因此极小极大法产生相异路径的效果不如 Yen 改进算法好。

4.2 计算时间比较

为比较传统 Yen 算法和 Yen 改进算法的计算效率，本文构建了包含 300 个节点的拓扑图，如图 4 所示。分别通过传统 Yen 算法和 Yen 改进算法计算相同

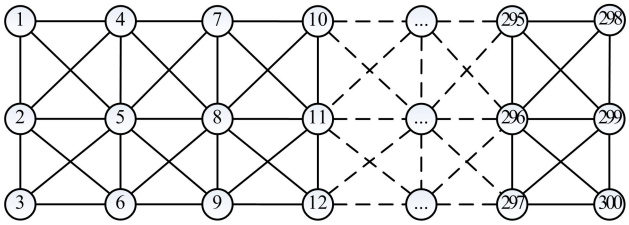

图 4 计算效率比较所用拓扑图

源节点到目的节点的路径，共进行 6 轮计算，每一轮的源节点均为 1，不同轮的目的节点数量分别为 49、99、149、199、249 和 299。每对源节点和目的节点计算 5 条最短路径，2 种算法计算所用的时间如表 4 所示。

表 4 传统 Yen 算法和 Yen 改进算法的计算时间对比情况

源节点	目的节点	计算时间/s	
		传统 Yen 算法	Yen 改进算法
①	[②,③,④,…,㊿]	3.6962827	2.112946
①	[②,③,④,…,㉚]	14.7518486	7.9388349
①	[②,③,④,…,㊿]	33.1470758	17.3526006
①	[②,③,④,…,㉚]	58.2971951	30.2884612
①	[②,③,④,…,㊿]	91.0461688	46.7941855
①	[②,③,④,…,㉚]	130.4922458	66.6491549

从表 4 可以看出，在求解目标一致的情况下，Yen 改进算法计算所用时间大概为传统 Yen 算法的一半。

4.3 算法时间复杂度分析

本文提出的 Yen 改进算法使用 Dijkstra 算法求解最短路径时，假设网络节点数为 n，边总数为 e，则实现 Dijkstra 算法的时间复杂度为 $O(n\log n)$。Yen 改进算法多次调用 Dijkstra 算法，单独计算完全节点偏离路径和完全路径偏离路径。每次调用前，对图 G 进行处理，如果有 m 个节点需要过滤，则进行 $n\times m$ 次处理；如有 r 条边进行处理，则进行 $e\times r$ 次处理。假设调用次数为 C，则 $C\geqslant K$，因此 Yen 改进算法的复杂度为 $O[C(n\log n+nm+er)]$，其复杂度在数量级上和其它改进算法基本持平。

5 结束语

为光网络用户业务规划多条光路径应用场景时，针对传统 Yen 算法计算路径相异性小的问题，本文引入节点完全偏离和路径完全偏离的思想，提出了一种 Yen 改进算法。测试结果表明：在拓扑图中存在平行边的情况下，Yen 改进算法计算的最短路径相异性明显

优于传统 Yen 算法。Yen 改进算法还采用启发式搜索方法，使得路径的整体搜索效率得到提高。此外，还构建了与实际生产环境类似的各类拓扑图，设计了不同的计算初始条件（如源、目的和路径计算数量等）进行路径计算，并记录测试数据，从路径相近性和路径计算时间这 2 个维度与极小极大法、传统 Yen 算法进行比较，证明了本文提出的 Yen 改进算法的有效性。

参考文献：

[1] 王刊良，徐寅峰. 相异路径选线问题的模型与算法[J]. 系统工程理论方法应用，2001，10(1)：9–10.
[2] STEPHEN L, DANNY K. 人工智能[M]. 林赐，译. 第二版. 北京：人民邮电出版社，2018.
[3] BRANDER A W, SIMCLAIR M C. Comparative study of K-shortest path algorithms [C]//John Moores University. Proc.11th UK Performance Engineering Workshop. Livepool: Springer London，1995: 370–379.
[4] 高松，陆峰. K 则最短路径算法效率与精度评估[J].中国图象图形学报，2009，14(8)：1677–1683.
[5] WALTER H, RICHARD P. A method for the solution of the Nth best path problem[J]. Journal of the ACM，1959，6(4)：506–514.
[6] NAOKI K, TOSHIHIDE I, HISASHI M. Anecient algorithm for K shortest simple paths[J]. Networks，1982. 12(4)：411–427.
[7] JIN Y Y. Finding the K shortest loopless paths in a network[J]. Management Science, 1971, 17(11): 712–716.
[8] LAWLER E L. A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem[J]. Management Science, 1972, 18(7): 401–405.
[9] MARTINS E Q V, SANTOS J L E. A new shortest paths ranking algorithm[J]. Inverstigacao Operacional, 2000, 20(1): 47–62.
[10] HERSHBERGER J, MAXEL M, SUR S. Finding the K shortest simple paths: A new algorithm and its implementation [J]. ACM Transactions on Algorithms，2007，3(4)：45–48.
[11] 柴登峰，张登荣. 前 N 条最短路径问题的算法及应用[J]. 浙江大学学报，2002，36(5)：531–534.
[12] 袁红涛，朱美正. K 优路径的一种求解算法与实现[J]. 计算机工程与应用，2004，40(6)：51–53.
[13] 高松，陆峰，段滢滢. 一种基于双向搜索的 K 则最优路径算法[J]. 武汉大学学报(信息科学版)，2008，33(4)：418–421.
[14] VEDAT A, ERHAN E, RAJAN B. On finding dissimilar paths [J]. European Journal of Operational Research, 2000, 121: 232–246.