K 最短路径算法综述

徐 涛[1,2]，丁晓璐[1,2]，李建伏[1]

（1. 中国民航大学 计算机科学与技术学院，天津 300300；2. 中国民航信息技术科研基地，天津 300300）

摘 要：为了进一步推广应用 K 最短路径（K shortest paths，KSP）算法并为深入研究该类算法提供相关资料。根据路径限制条件，将 KSP 问题分为一般 KSP 问题和限定无环 KSP 问题，归纳总结了求解每类 KSP 问题的基本思路、研究现状和研究进展。KSP 问题非常复杂，在实际应用中所需处理的数据规模非常庞大，使得算法效率成了评价 KSP 算法的一个重要指标。在分析各种 KSP 算法时尤其关注其时间复杂度，指出 KSP 问题未来的研究方向，将为满足多约束的最短路径等问题的研究提供有益的参考。

关键词：KSP 问题；路径限制条件；一般 KSP 问题；限定无环 KSP 问题；时间复杂度

中图法分类号：TP301 **文献标识号**：A **文章编号**：1000-7024（2013）11-3900-07

Review on K shortest paths algorithms

XU Tao[1,2], DING Xiao-lu[1,2], LI Jian-fu[1]

(1. College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;
2. Information Technology Research Base, Civil Aviation Administration of China, Tianjin 300300, China)

Abstract: To promote the applications of K shortest paths algorithm (KSP), and to provide the relevant information for the further research on this algorithm, a review on the recent progress of KSP algorithms is given. According to restriction constrained on paths, the KSP problem can be classified into two types: the general KSP and the constrained loopless KSP. The basic ideas, research status and research progress for solving each type of two KSP problems are reviewed. Since KSP problem is very complex and the scale of the graph in application is very large, the efficiency has become an important indicator to evaluate a KSP algorithm. Time complexity is consequently a special concern in analyzing all kinds of KSP algorithms. Finally, future research directions of KSP are pointed out. The above work can give a valuable reference for multiple constraint shortest path problems.

Key words: K shortest paths problem; restriction condition; general K shortest paths problem; constrained loopless K shortest paths problem; time complexity

0 引 言

K 最短路径问题（K shortest paths，KSP）是最短路径问题的一种变形。与最短路径问题不同，KSP 问题[1]的目的是寻找图中起点和终点间的多个备选优化路径，形成最短路径组，以最大程度满足用户对不同路径的选择需求。

KSP 问题的研究对于满足多约束的最短路径问题等其它最短路径问题的研究具有重要意义。除此之外，KSP 问题具有广阔的应用背景[2,3]。例如在交通查询系统中需要给用户提供多条可供选择的最短路径。另外，QoS 路由问题、机器人动作规划问题、计算分子生物学中的序列比对问题、有限长度的 Huffman 编码问题等都可以看作是 KSP 问题。

KSP 问题最早由 Hoffman 和 Pavley 在 20 世纪 50 年代提出，多年来一直受到业界的广泛关注，现有已经提出了各种求解算法。根据路径限制条件，KSP 问题可以被分为两种：限定无环 KSP 问题和一般 KSP 问题。前者要求所有求得的路径都必须是简单路径，不能含有环；后者则对路径没有任何限制。两类 KSP 问题具有不同的特征，目前针

收稿日期：2013-02-01；**修订日期**：2013-04-11

基金项目：天津市应用基础及前沿技术研究计划基金项目（09JCYBJC02300）；中央高校基本科研业务费用中国民航大学专项 B 类基金项目（ZXH2011B003）

作者简介：徐涛（1962-），男，重庆人，教授，博士生导师，研究方向为民航信息系统理论、智能信息处理；丁晓璐（1988-），女，河南濮阳人，硕士研究生，研究方向为民航信息系统及应用；李建伏（1979-），女，河北沧州人，博士，讲师，研究方向为民航信息系统理论、人工智能。E-mail: dingxiaolu88@163.com

对每种KSP问题分别提出了不同的求解算法，分别称之为限定无环KSP算法和一般的KSP算法。本文将从算法的基本思想、研究现状和研究进展等方面对每类算法进行归纳总结。由于KSP问题的时间复杂度非常高，加上在实际应用中所要处理的图非常庞大，时间复杂度成了评价KSP算法的重要指标。因此，在分析算法时尤其关注算法的时间复杂度。

1 KSP问题

假定$G=(V, E)$表示一个图，其中V为n个节点的集合，E为m条边的集合。当E中每条边都为有向边时，G为有向图；当每条边都为无向边时，G为无向图。不失一般性，这里假定图G为有向图（可以通过将无向图中的每条边看作是两条相反方向的边将其转化为有向图）。E中每条边e_k用一个节点对表示，即$e_k=(i, j)$，则称i为e_k的尾节点，j为e_k的头节点。用$In(i)=\{(j, i): (j, i)\in E, j\in V\}$表示头节点为$i$的所有边的集合；$Out(i)=\{(i, j): (i, j)\in E, j\in V\}$表示尾节点为$i$的所有边的集合。每条边$e_k=(i, j)$与某个$c_{i,j}$相关，表示该边的长度。

假定s和t是图G中的两个节点。图中从s到t的路径p由节点序列表示，即$p=(v_1=s, v_2, \cdots, v_h=t)$，其中，$1<h\leqslant n$，并且对于所有的$j=1, \cdots, h-1$，都有$(i_j, i_{j+1})\in E$。$s$和$t$分别称为$p$的初始节点和终止节点。$p$的长度$c(p)$为$p$上所有边的长度之和，即$c(p)=\sum_{(i,j)\in p}c_{ij}$。

当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。环是从某个节点到其自身的路径，其中除了初始节点与终止节点相同外，其它节点都不相同。为了确保存在解，KSP问题一般都要求图中不存在负环。所谓负环是指长度小于0的环。

从s到t的路径集合用P_{st}表示。最短路径问题是要找到图G中从s到t的具有最小长度的路径$p*$，即确定$p*\in P_{st}$，使得对于任何其它$p(p\in P_{st}, p\neq p*)$都有$c(p*)\leqslant c(p)$。KSP问题是对最短路径问题的推广，它除了要确定最短路径之外，还要确定次短路径、第三短路径，$\cdots$，直到找到第K短路径为止。用p_i表示从s到t的第i短路径，KSP问题是确定路径集合$P_K=\{p_1, p_2, \cdots, p_K\}\in P_{st}$，使得满足以下3个条件：

(1) K条路径是按次序产生的，即对于所有的i（$i=1, 2, \cdots, K-1$），p_i是在p_{i+1}之前确定的；

(2) K条路径是按长度由小到大排列的，即对于所有的$i(i=1, 2, \cdots, K-1)$，都有$c(p_i)<c(p_{i+1})$；

(3) K条路径是最短的，即对于所有的$p\in P_{st}-P_K$，都有$c(p_K)<c(p)$。

根据路径限制条件，KSP问题通常被分为两种：一般KSP问题和限定无环KSP问题。前者对路径没有任何限制，如上述对KSP问题的定义；后者则要求所求得的路径都必须是简单路径，不能含有环。因此，限定无环的KSP问题是在一般KSP问题的基础上增加第4个条件，即所求得的K条路径都是无环的。

由于一般KSP问题和限定无环的KSP问题具有不同的特征，用于求解每种KSP问题的算法也不相同，分别称之为限定无环KSP算法和一般的KSP算法。目前限定无环KSP算法主要有偏离路径算法与改进Dijkstra算法，一般KSP算法有标号算法、删除路径算法、偏离路径算法与改进智能算法4种。KSP算法的分类如图1所示。

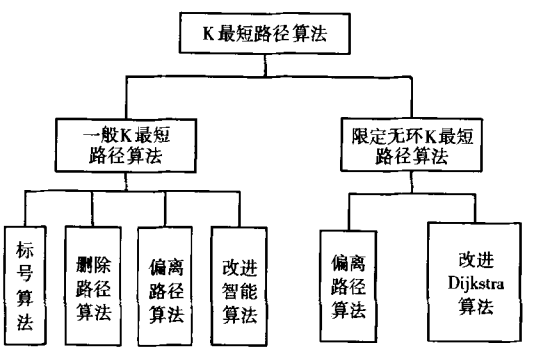

图1 KSP算法分类

2 限定无环KSP算法

目前限定无环KSP算法主要有偏离路径算法与改进Dijkstra算法。

2.1 偏离路径算法

这类算法的目标是要构建包含K个最短路径的K最短路径树T_K，树T_K的根节点为s，叶子节点为终节点t的K个备份（因此，这种树与实际意义上的树不同，因为其中包含重复节点）。树中每个从s到叶子节点的路径为K条最短路径之一。假定图G中从$s=1$到$t=5$的前3条最短路径$p_1\sim p_3$分别为$p_1=\{1, 2, 5\}$，$p_2=\{1, 3, 4, 5\}$，$p_3=\{1, 2, 4, 5\}$，该3条路径形成的最短路径树T_3如图2所示，其中每个叶子节点旁边标注的数字表示从根节点到该叶子节点的路径的长度，如路径p_3长度为2。

最短路径树T_K的构建基于一个重要概念——偏离路径。假定存在从s到t的两条路径$p=(v_1, v_2, \cdots, v_l)$和$q=(u_1, u_2, \cdots, u_w)$，如果存在一个整数$x$满足以下4个条件：

(1) $x<l$，并且$x<w$；

(2) $v_i=u_i$ $(0\leqslant i\leqslant x)$；

(3) $v_{x+1}\neq u_{x+1}$；

(4) $(u_{x+1}, u_{x+2}, \cdots, u_w=t)$是从$u_{x+1}$到$t$的最短路径。

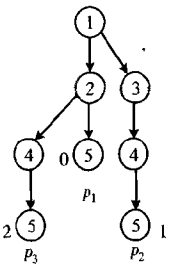

图 2 K 最短路径树 T_3

则称 (u_x, u_{x+1}) 为 q 相对于 p 的偏离边，u_x 为 q 相对于 p 的偏离节点，路径 $(u_{x+1}, u_{x+2}, \cdots, u_w=t)$ 为 q 相对于 p 的最短偏离路径。如图 2 中，p_2 相对于 p_1 的偏离节点为节点 1，偏离边为边 (1, 3)，偏离路径为 (1, 3, 4, 5)。

偏离路径算法的核心在于利用已经求得的 p_1，p_2，…，p_k 的最短偏离路径产生 p_{k+1}。现在已经提出了多种偏离路径算法，各种算法的不同之处在于偏离路径的计算和偏离路径的获取方法。

最早的偏离路径算法由 Yen 在 1971 年提出。Yen 算法首先使用标准的最短路径算法（如 Dijkstra 算法）找到从 s 到 t 的最短路径，将其作为 p_1 并放入结果列表 A 中。在求得了前 k 条路径 $\{p_1, p_2, \cdots, p_k\}$ 之后，计算 p_{k+1} 的过程如下：

(1) 取 p_k 中除了终止节点 t 之外的每个节点 v_i 作为可能的偏离节点，计算 v_i 到节点 t 的最短路径。在计算 v_i 到 t 的最短路径时，需要满足以下两个条件：第一，为了保证无环，该路径不能通过当前最短路径 p_k 上从 s 到 v_i 之间的任何节点；第二，为了避免与以前找到的路径重复，从节点 v_i 分出的边不能与以前找到的最短路径 p_1，p_2，…，p_k 上从 v_i 分出的边相同。

(2) 在找到了 v_i 与 t 之间满足以上两个条件的最短路径后，将该最短路径与当前路径 p_k 上从 s 到 v_i 的路径拼接在一起构成 p_{k+1} 的一条候选路径，并将其存储在候选路径列表 B 中。

(3) 从候选路径列表 B 中选择最短的一条作为 p_{k+1}，并将其放入结果列表 A 中。以上过程不断重复，直到得到 K 条路径为止。

Yen 算法的时间花费主要体现在以下四点：第一，最短路径的计算，目前 Dijkstra 算法求解两点间最短路径的时间花费为 $O(m+n\log n)$；第二，考虑到在结果列表 A 中最多有 K 条无环路径需要维护，每次插入新的候选路径需要的时间为 $O(\log K)$；第三，每条路径 p_{k+1} 最多包含 n 个节点，因此求每个节点到目的节点的最短路径需要的时间花费为 $O(n*(m+n\log n))$；第四，p_{k+1} 最多有 n 个候选路径，从中选择最短的需要时间为 $O(n)$。因此，Yen 算法的复杂度是 $O(c(n)+K(n*(m+n\log n)+n+\log K))$，即 $O(Kn(m+n\log n))$。

Yen 算法每次求 p_{k+1} 时，将 p_k 上除了终止节点之外的所有节点都视为潜在的偏离节点进行计算，当该节点不在 p_k 的偏离路径上时就会产生大量重复路径，Lawler 对其进行了改进，只将在 p_k 的偏离路径上的节点作为潜在的偏离节点进行计算。虽然在最坏情况下，Lawler 算法与 Yen 算法具有相同的时间复杂度，但研究表明 Lawler 算法的实际效率要高于 Yen 算法。

Martins 等对 Yen 算法进行改进提出了 MPS 算法。该算法引入了一个量——边的缩小长度（reduced cost）。用 π_x 表示从 x 到 t 的最短路径长度，边 (i, j) 的缩小长度定义为 $\bar{c}_{i,j}=c_{ij}+\pi_j-\pi_i$，并且满足以下 3 个特征：

(1) 对于所有的边 (i, j) $\in$E，都有 $\bar{c}_{i,j}\geqslant 0$；

(2) 对于所有的边 (i, j) $\in$Tt (Tt 为图 G 中每个节点 x 到 t 的最短路径树)，都有 $\bar{c}_{i,j}=0$；

(3) $\bar{c}(q)=\sum_{i=1}^{w}\bar{c}_{u_{i-1},u_{i-1}}=\pi_t-\pi_s+c(q)$。

最后一个特征说明按 c 值与按 $\bar{c}$ 值对路径进行排序的结果是一样的。另外，由第二个特征可以导出 $\bar{c}(q)=\bar{c}(p)+\bar{c}_{u_x,u_{x+1}}$，其中 (u_x, u_{x+1}) 为 q 相对于 p 的偏离边。这说明，从 p 产生的下一条最短路径的各种候选路径的 $\bar{c}$ 值的差别只在于偏离边的 $\bar{c}$ 值的差别。因此，在生成候选路径时，没有必要计算每条候选路径的长度值，而只需计算各候选路径的偏离边的 $\bar{c}$ 值，最后选择偏离边 $\bar{c}$ 值最小的候选路径作为下一个最短路径即可，从而简化了偏离路径长度的计算。虽然最坏情况下，MPS 算法和 Yen 算法具有一样的时间复杂度。但是作者对随机网络的测试结果表明在实际应用中 MPS 的运行速度要比 Yen 快。

Martins 等提出了关于 Yen 算法的一种新的实现方法。如上所述，Yen 算法在计算节点 v_i 到节点 t 的最短路径时，为了使得路径无环，该路径不能通过当前最短路径上从 s 到 v_i 之间的任何节点；为了避免与以前找到的路径重复，从 v_i 分出的边不能与以前找到的最短路径上从该节点分出的边相同。为了满足以上条件，Yen 算法在实现时要从当前网络 G 中删除当前最短路径 p_k 上从 s 到 v_i 的所有节点和出现在 T_k 上的所有的 $Out(v_i)$。在修改后的网络中求得 v_i 到 t 的最短路径之后再恢复所有被删除的边和节点。这样，Yen 算法就会频繁地改变图 G，有多少个可能的偏离节点 v_i 就会改变多少次。Martins 提出的新实现方法将图的改变次数降低为每求一个最短路径 p_{k+1} 改变一次。与 Yen 算法的区别在于分析 p_k 上潜在的偏离节点 v_i 的次序不同：Yen 算法采用顺序的次序进行分析，即先分析 v_i，再分析 v_{i+1}，…直到 p_k 中除终止节点 t 之外的最后一个节点 v_h，每分析一个节点就要改变网络一次；Martins 则采用倒序，即首先分析 v_h，再分析 v_{h-1}，…。在分析 v_h 时，与 Yen 算法一样，要删除 G 中当前最短路径 p_k 上从 s 到 v_h 的所有节

点和出现在 T_k 上的所有的 $Out(v_h)$。这相当于一次将 p_k 整条路径从图 G 中删除，在分析完 v_h 后在图中恢复该节点。在分析 v_{h-1} 时，由于当前最短路径 p_k 上从 s 到 v_{h-1} 的所有节点已经在分析 v_h 时被删除，因此，没有必要再进行删除操作，网络的结构也不会发生变化。虽然在最坏情况下，Martins 算法和 Yen 算法具有一样的时间复杂度，但作者通过对包含 5000 和 10000 个节点的随机网络的测试结果表明 Martins 算法的运行效率要高。

J. Hershberger[4]等在 Yen 算法的基础上提出了一个偏离路径算法。与 Yen 算法的不同点在于，Hershberger 将候选路径划分为多个等价类，用替代路径算法来求得每个等价类的最短路径并存放入堆中，最后将堆中长度最小的路径作为下一个最短路径。该算法最坏时时间复杂度与 Yen 一样，但在理想情况下能达到 $O(K(m+n\log n))$。作者对具有 500 个节点与 12000 条边的 GIS 数据的测试结果显示 J. Hershberger 的算法要比 Yen 的算法快 4－8 倍；对模拟无线网的模拟生成数据的测试结果显示 J. Hershberger 的算法要比 Yen 的算法快 20 倍。

2.2 改进 Dijkstra 算法

Dijkstra 算法是典型的最短路径算法，用于计算一个节点到其它所有节点的最短路径。其特点是以起点为中心向外层逐步扩展，直到扩展到终点为止，能够精确的求出一条最短路径。因此，许多 KSP 问题的研究者提出了多种不同的基于 Dijkstra 算法求解 K 最短路径的改进算法。

柴登峰与张登荣[5]设计了一个递归调用 Dijkstra 算法的新算法，进行 N 次循环，第 i 次循环确定第 i 条最短路径，循环中先选择地 $i-1$ 最短路径，根据已产生的 $i-1$ 条路径产生若干子图，求取其上的第 1 最短路径作为候选路径存放入路径集合中。每次存入新候选路径时采用排序算法插入到正确位置，这样路径集合中的元素始终保持按路径长度大小进行存放。其时间复杂度是 $O(e*n^2)$，其中 e 与 n 分别是图中边和顶点的数目

利用 Dijkstra 算法求解 K 条最优路径时，需要多次调用该算法，使得算法时间效率较低。袁红涛[6]等分析了 Dijkstra 算法求解最短路径时，每次更新完当前点的所有邻接点权值后，对所有未标识的点按照所有未标识的点到起点的权值进行快速排序，而这个过程并不是最优的。因此，对 Dijkstra 算法进行改进，其处理过程如下：在更新每个未标识点到起点的权值时对该节点进行折半查找插入排序，对于每个节点最差情况下也只需要进行 n 次比较（n 远远小于节点数目），而全部重排的代价是（以平均性能最优的快速排序为例）平均时间为 $k*n\log n$（k 是常数因子，n 是待排序的个数≤节点数），由此可以看出折半查找插入排序比快速排序更优。

高松[7]等在 2008 年提出了一种基于双向搜索策略的 K 则最优路径算法，以改进的 Dijkstra 最优路径算法为基础，从起点和终点同时搜索，分别构造正序和逆序最优路径树，计算网络中两点之间的多条参考 K 则最优路径。其时间复杂度是 $O(2n+2n+k^2m+km+n\log n)$，由于 m 的上限为 n，因此在最坏情况下该算法的时间复杂度为 $O(k^2n+n\log n)$。

白轶多[8]等也在采用 Dijkstra 算法求出最短路径的基础上，提出了一种基于前 $k-1$ 条最短路径的 k 次短路径的求解算法。

3 一般 KSP 算法

一般 KSP 问题与限定无环 KSP 问题的区别在于一般 KSP 问题对路径没有任何限制。因此，可以将求解限定无环 KSP 问题的偏离路径算法进行改进用以求解一般 KSP 问题。另外，研究表明，一般 KSP 问题满足最优化原理（optimality principle）。对于 KSP 问题，最优化原理认为对于任意 $k\geqslant 1$，从 s 到 t 的第 k 短路径 p_{st}^k 是由每个从 s 到 p_{st}^k 的每个中间节点 i 的第 $\varepsilon_i(1\leqslant\varepsilon_i\leqslant k)$ 个最短子路径构成的，即对于 p_{st}^k 上的节点 i，有 $p_{st}^k=p_{si}^1\cup p_{it}^{k-1}$ 或…或 $p_{si}^2\cup p_{it}^{k-1}$。现在已经开发出两种基于最优化原理的一般 KSP 算法，即标号算法和删除路径算法。

3.1 标号算法

求解 KSP 问题的标号算法通常是对求解最短路径的标号算法的扩展。与求解最短路径问题的标号算法一样，求解一般 KSP 问题的标号算法也分为标号纠正（labeling correcting）算法和标号设置（labeling setting）算法两类。

标号纠正算法的基本思想是：将起始节点到每个节点的唯一标号（这里标号指的是起始节点到每个节点的距离值）以多重标号链表表示，即对每个节点建立一个标号链表。在每次迭代过程中，选择某个节点并更新与该节点相关联节点的标号链表。以上过程直到不再有更新为止。

最早标号纠正算法是 Shier 提出的双向扫除（double-sweep algorithm）算法。其基本思想是算法通过多次迭代完成，每次迭代由前向扫除和后向扫除两个过程组成：

（1）前向扫除：按照节点编号的递增顺序，连续地检索所有与 j 节点相邻的前一个节点 $i(i<j)$，确定从起始节点到 j 节点的 K 条最短路径是否可能通过这个相邻的节点。若这样的路径存在，则取新的估计值，并用于下一步迭代。

（2）后向扫除：执行类似的算法，其过程是按照节点编号递减的序列，检查与 j 顶点相邻的后一个节点 i（按图的方向），即 $i>j$。按照以上过程不断迭代，直到每个节点的每个标号值不再改变为止。

双向扫除算法最多需要 $nk/2$ 次双向扫描，每次双向扫除需要 n^2 次推广的极小运算和推广的加法运算，每次推广的加法运算最多需要 $k(k-1)$ 次加法和比较，因此双向扫除算法的时间复杂度为 $O(nk*n^2*k(k-1)/2)$，即 $O(n^3k^3)$。

K. A. Rink 等在 2000 年对双向扫除算法的计算过程进

行了简化，使其时间复杂度降为 $O(n^3k^2)$。

F. Guerriero 提出了一类新的标号纠正算法。该算法包括 K 次不同的扫除。在第一次扫除结束之后，计算最短路径的长度；第二次扫除结束后计算次短路径的长度，依此类推。在第 k 次扫除时，算法不断地提高第 k，$k+1$，…，K 短路径长度的当前估计。值得指出的是这个方法能够充分利用以前产生的信息。实际上，当从 s 到 t 的第 k 短路径的长度的上限被提高后，不是丢弃它，而是立即用它来更新后续最短路径的标号。作者通过在每次迭代中采用不同的节点选择策略与双扫除算法进行了比较，结果表明该类算法的运行时间与双扫除算法相当，甚至对于某些实现策略，该类算法比双扫除算法更快。

标号设置算法与标号纠正算法类似，不同点在于在每次迭代过程中，选择某个节点并更新与该节点相关联节点的标号列表并把这个标号作为永久标号。以上过程直到找到 K 条路径为止。

最早的标号设置算法由 Dreyfus 提出，该算法的时间复杂度为 $O(kn^2)$。之后 Fox 将求解最短路径问题的经典的标号设置算法——Dijkstra 算法扩展到了 KSP 问题中，算法能在 $O(m+Kn\log(n))$ 内求得从源节点到图中每个节点的 K 条最短路径。2009 年，Paluch[9] 提出了一种新的多标号算法来计算 K 条最短路径问题，该算法能够直接应用于有向图中，也能够通过将无向图转化为有向图的方法应用与无向图中。

3.2 删除路径算法

第一个删除算法是由 Martins 在 1984 年提出。该类算法基于以下特征：某特定两点在图 G 中的第二最短路径 p_2 是该两点在图 G' 中的最短路径，其中图 G' 是由图 G 通过删除最短路径 p_1 得到的；第三最短路径是该两点在删除 p_1 和 p_2 后得到的图 G'' 中的最短路径。因此，删除路径算法一般包括以下两步：①从当前的图中删除某最短路径；②确定新产生图的最短路径。这类算法的关键在于第(1)步。

从图 G 中删除路径 $p=<s=v_0, v_1, v_2, \cdots, v_l=t>$ 产生图 G' 的过程一般包括以下四步：

(1) 为路径 p 的每个中间节点 $v_i(1<i<l)$ 建立一个备份节点 v_i'，产生新节点集合 $V'=V\cup\{v_1', v_2', \cdots, v_{l-1}'\}$。通过以上描述可知没有建立 v_0 的备份节点 v_0'，但为了便于描述，下文中也会用到 v_0'，此时 v_0' 与 v_0 都表示同一个节点 v0；

(2) 将节点 $\{(v_{i-1}', v_i')\}$ $(1<i<l)$ 连接起来；

(3) 将 v_i 的不在路径 p 上的前驱节点与每个节点 v_i' 相连，即 v_i' 的入弧 $in(v_i')$ 为

$$in(v_i')=\{(j,v_i')\mid(j,v_i)\in E, j\in V-\{v_{i-1}\}\}\cup\{(v_{i-1}',v_i')\}$$

(4) 将弧 (v_{l-1}, v_l) 移到 (v_{l-1}', v_l)（因此，路径 $p=<s=v_0, v_1, v_2, \cdots, v_l=t>$ 从 G' 中删除了）。

用 $T_s(T_s')$ 表示图 $G(G')$ 的以 s 为根节点的最短路径树，如果 T_s 已知，很容易就能得到 T_s'。实际上，每一个节点 $v\in V-\{t\}$ 的标号（从 s 节点到 v 节点的距离）在 T_s' 中不变。新产生的节点 v_i' 的标号为

$$\pi_{v_i'}=\min\{\pi_j+c(j,v_i'):(j,v_i')\in A(v_i')\}$$

其中，π_j 为图 G 中从 s 到 j 的最短路径的长度。

可以证明，图 G' 中从 s 到 t 路径的集合对应于集合 $P-\{p\}$。当 $|in(v_i')|=1$ 时，则在图 G' 中 v_i' 是多余的，因为没有弧进入这个节点。因此，提出了很多改进算法以去掉冗余节点，从而产生了使得从 s 到 v_i' 的最短路径对应于从 s 到 v_i 的下一条最短路径的算法。

Martins 和 Santos 注意到节点 v_i' 产生之后，$In(v_i')$ 就再也不会被用到，因此他们将 $in(v_i')$ 的信息存放在 $In(v_i)$ 中，提出了 MS 算法，使得算法的空间复杂度从 $O(K(n+m))$ 降为 $O(Kn+m)$，时间复杂度为 $O(Km)$ 与原算法保持不变。

Martins 随后又改进了 MS 算法，将 $In(v_i)$ 的尾节点按其标号值进行排序，使算法的时间复杂度降为 $O(m+Kn\log n)$，空间复杂度与 MS 算法一样，都为 $O(Kn+m)$。

3.3 偏离路径算法

针对一般 KSP 问题的偏离路径算法的研究始于 Eppstein 提出的 EA 算法。EA 算法基于路径的一种特殊表示。即假定图 G 中所有节点到节点 t 的最短路径形成的最短路径树为 T。此时，图 G 中属于 T 的边被称为树边，否则被称为非树边。EA 算法将任何从 s 到 t 的路径 p 用 p 中的非树边序列 $\xi(p)$ 表示。如图 3 中用实线表示的边为树边，虚线表示的边为非树边，则路径 $p=\{s, c, a, f, b, t\}$ 可以用非树边序列 $\{(s, c), (a, f)\}$ 表示。

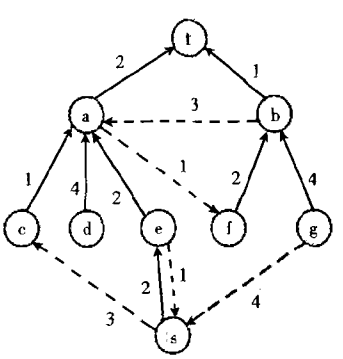

图 3 图 G 中的树边和非树边

由边 (i, j) 的缩小长度的定义 $\bar{c}_{i,j}=c_{ij}+\pi_j-\pi_i$ 导出每条从 s 到 t 路径 p 的长度等于从 s 到 t 的最短路径的长度与 p 上所有非树边的缩小长度之和。这样，为了找到从 s 到 t 的 k 条最短路径，只需找到非树边的缩小长度之和最小的 k 条路径。

为此，EA 构建了一个称为路径图的 $P(G)$ 来存放图 G 中所有的非树边。$P(G)$ 是一个有向加权图，每个节点

对应于 G 中的非树边（每个非树边至少对应于 $P(G)$ 的一个节点）。这些节点在 $P(G)$ 中以堆的形式进行排列。$P(G)$ 的结构确保了 $P(G)$ 中从根节点到任何节点之间的路径与 G 中从 s 到 t 的路径的偏离边序列表示一一对应。并且，$P(G)$ 路径越短，则与之对应的 G 中的从 s 到 t 的路径越短。

构建路径图 $P(G)$ 的过程如下：首先将 $P(G)$ 初始化为空。然后将图 G 中每个节点 v 的堆结构 $H(v)$ 加入到图 $P(G)$ 中。$H(v)$ 表示尾节点在图 G 中从 v 到 t 的最短路径上的非树边的集合，这些边按缩小长度值由小到大进行堆排序。$H(v)$ 中的每条边被看作是 $P(G)$ 的一个节点，并为每个节点 α 建立一条边指向与其相邻的下一个节点 β，该边长度为 β 节点所对应的非树边的缩小长度值与 α 节点所对应的非树边的缩小长度值之差。当几个节点堆结构相同时，则将其压缩为一个堆结构。接着为 $P(G)$ 中每个表示非树边 (u, v) 的节点 α 建立一个边指向 $H(v)$ 的根节点。这些边长度为 $H(v)$ 的根节点所对应的非树边的缩小长度值。最后为 $P(G)$ 增加一个新的节点 $*$，并建立一条边将其与 $H(s)$ 的根节点相连，该边的长度为 $H(s)$ 的根节点的缩小长度值。由图 3 得到路径图 $P(G)$ 如图 4 所示，其中每个节点由两部分组成，即对应的图 G 中的非树边以及该树边的缩小长度值。

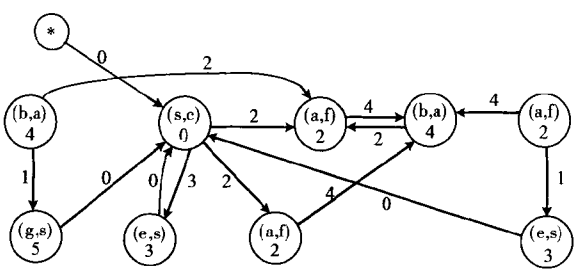

图 4　路径图 $P(G)$

最后对采用 Dijkstra 算法求得 $P(G)$ 中与根节点距离最短的 K 条路径，从而得到了 G 中从 s 到 t 的 K 条最短路径。

构建路径图图 $P(G)$ 需要的时间为 $O(m+nlogn)$，后来 Eppstein 又将其提高到 $O(m+n)$。运用 Dijkstra 求 $P(G)$ 中与根节点距离最短的 K 条路径需要的时间为 $O(K\log K)$。因此，在计算了图中所有节点到节点 t 的最短路径后，EA 算法求得 K 条最短路径花费的时间为 $O(m+n+K\log K)$。

V. M. Jimenez 等注意到 EA 算法的大部分时间花费在 $P(G)$ 的建立（需要的时间为 $O(m+n)$），而一旦 $P(G)$ 建立，就可以很快地（$O(K\log K)$）求得 K 条最短路径。因此，在 2003 年提出了一个 lazy 版本的 EA 算法，该算法只是建立选择 K 条路径确实需要的部分 $P(G)$ 图。理论上，与 EA 算法具有相同的时间复杂度，V. M. Jimenez 通过对各种随机图进行了大量的实验，结果表明，在实际应用中 V. M. Jimenez 的算法要比 EA 快得多。

Martins 在 EA 算法的基础上，采用了边的缩小代价来简化偏离路径距离的计算。该算法通过将每个偏离节点的所有潜在偏离边按缩小代价值 $\bar{c}$ 值递增的顺序进行排序，最后只需计算其中 $\bar{c}$ 最小的一条的偏离路径。改进算法的时间复杂度为 $O(m\log n+Kn)$。

Husain Aljazzar 等对 EA 进行改进提出了 K * 算法[10]。K * 算法的基本思想是首先用 A * 算法对图 G 进行搜索直到目标节点 t 被找到，接着根据 A * 搜索过程中遍历的边和扩展的节点建立 $P(G)$ 图，然后启动 Dijkstra 算法计算 $P(G)$ 中从根节点到其它节点的最短路径。如果 Dijkstra 算法能够得到 K 条最短路径，则算法成功结束；否则重新启用 A * 搜索继续对 G 进行搜索，接着是建立 $P(G)$ 图、启动 Dijkstra 算法对 $P(G)$ 进行搜索。重复以上过程，直到得到 K 条最短路径为止。K * 算法的时间复杂度为 $O(m+n\log n+K)$。

V. M. Jimenez 等提出了另一个偏离路径算法 REA(recursive enumeration algorithm)，该算法递归地访问第 $k-1$ 个最短路径中的每个节点来构建第 k 个最短路径，并为每个节点建立一个堆结构，用来存放从源节点到该节点的最短路径的候选路径。在计算了图中所有节点到节点 t 的最短路径后，REA 算法求得 K 条最短路径所需要的总时间为 $O(m+Kn\log(m/n))$。此外，Roditty[11] 提出了一种新的近似算法来求解加权有向图中的 K 条简单最短路径，其时间复杂度是 $O(k(mn+n^{3/2}\log n))$。

3.4　改进智能算法

近年来，许多学者对于基于进化和仿生计算的智能算法求解复杂优化问题进行了深入研究，获得了许多理论和应用成果。同时，也为求解 K 最短路径问题提供了另一种可行途径。

马炫[12]在 2006 年提出了任意两点间 k 条最优路径问题的遗传算法，采用节点的自然路径作为染色体编码，根据路径节点的连接实施染色体的交叉操作，将节点路径块作为染色体的变异基因块实施变异操作，使得染色体的编码形式与节点自然路径的表示相一致，从而实现了问题空间与遗传搜索空间的统一，适用于大规模网络中求解任意两点间的多条路径。此外，黄泽汉[13]等人针对网络优化调度问题，采用以“时间换空间”的思想，在传统的蚁群算法基础上，设计了一种基于时间扩展的网络 K-最短路径算法。

2011 年，马炫与刘庆[14]又提出了一种求解 k 条最短路径问题的混合蛙跳算法。采用自然路径的形式对青蛙个体编码，设计了一种能够使模因信息在青蛙个体间传递的蛙跳方法。在青蛙族群内部，通过较差个体向优秀个体的跳跃进行局部搜索，从而优化模因信息。在族群直接，通过混合与排序使各族群的模因信息得以交流与重组，从而获

取新的寻优方向，由此实现对 k 条最短路径问题的求解。

林洁[15]等针对智能交通系统中的求解 K 条备选最优路径的问题，借鉴智能优化搜索算法的思想和生物免疫系统的隐喻机制，提出了一种基于人工免疫优化搜索算法。在一般免疫优化算法的框架基础上，建立了一种新型人工免疫优化算法来求解 K 路最短问题。算法引进了生物免疫系统抗体对抗原的识别、免疫记忆、克隆选择和阴性选择等自然免疫机制，并通过疫苗的提取与接种加速算法的收敛。算法的实验结果可以看出，随着网络规模的扩大，人工免疫优化算法在动态网络最短路径搜索的实时性等方面将更加优于 Dijkstra 等常规算法和常用的遗传算法。

4 结束语

本文给出了与 KSP 问题相关的定义与概念，根据对求得的 K 条路径有无限制条件，将 KSP 问题划分为一般 KSP 问题和限定无环的 KSP 问题，并阐述了两类问题的常用求解方法和研究进展。

KSP 问题与最短路径问题同时出现在 20 世纪 50 年代，最短路径现在已经拥有许多成熟的算法，而 KSP 问题没有一个算法是被公众认可的。并且目前对 KSP 问题的研究更多的是从应用的角度，对算法本身研究的少，如 Eppstein 所建立的关于 KSP 算法的在线目录的几百篇文献中，只有 50 多篇是关于算法研究的。KSP 问题研究滞后的原因在于，KSP 问题极其复杂，加上在实际应用中处理的数据规模庞大，这给研究带来了很大的困难。随着计算能力的增强，以及越来越有效的数据结构的出现，加上现在一些实际应用迫切需要新的有效的 KSP 算法，在不久的将来 KSP 问题将会迎来其研究高潮时期。将来 KSP 问题的研究主要有以下几个方面：

新 KSP 算法：目前求解 KSP 问题的方法比较单一，如对于限定无环问题，现有的各种算法都是对 Yen 算法或 Dijkstra 算法的各种改进。因此，突破传统算法的新 KSP 算法的研究将会成为 KSP 问题研究的一个重要方向。

并行化：KSP 问题复杂度高，是一个典型的 NP 完全问题。并行化无疑是一个降低算法时间复杂度的好方法，现在关于 KSP 算法并行化的文献还很少。因此，KSP 算法的并行化也是一个研究方向。

应用：KSP 问题具有很强的应用背景，如何求解一些实际的 KSP 问题是将来 KSP 问题研究又一热点。

参考文献：

[1] GAO Song, LU Feng. The Kth shortest path algorithms: Accuracy and efficiency evaluation [J]. Journal of Image and Graphics, 2009, 14 (8): 1677-1683 (in Chinese). [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.]

[2] Androutsopoulos K N, Zografos K G. Solving the k-shortest path problem with time windows in a time varying network [J]. Operations Research Letters, 2008, 36 (6): 692-695.

[3] Wang Z P, Li G, Ren J W. A new search algorithm for transmission section based on k shortest paths [J]. Transactions of China Electrotechnical Society, 2012, 27 (4): 193-201.

[4] 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.

[5] CHAI Dengfeng, ZHANG Dengrong. Algorithm and its application to N shortest paths problem [J]. Journal of Zhejiang University, 2002, 36 (5): 531-534 (in Chinese). [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.]

[6] YUAN Hongtao, ZHU Meizheng. A fast algorithm and its implementation of finding the K-shortest paths [J]. Computer Engineering and Applications, 2004, 40 (6): 51-53 (in Chinese). [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.]

[7] GAO Song, LU Feng, DUAN Yingying. A Kth shortest path algorithm implemented with bi-directional search [J]. Geometrics and Information Science of Wuhan University, 2008, 33 (4): 418-421 (in Chinese). [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.]

[8] BAI Yiduo, HU Peng, XIA Lanfang, et al. A kth-shortest path algorithm based on k-1 shortest paths [J]. Geomatics and Information Science of Wuhan University, 2009, 34 (4): 492-494 (in Chinese). [白铁多，胡鹏，夏兰芳，等. 关于 k 次短路径问题的分析与求解 [J]. 武汉大学学报，2009，34 (4)：492-494.]

[9] Paluch S. A multilable algorithm for k shortest paths problem [J]. Komunikacie, 2009, 11 (3): 11-14.

[10] Aljazzar H, Leue S. K*: A directed on-the-fly algorithm foe finding the k shortest paths [R]. Technical Report soft-08-03, Chair for Software Engineering, University of Konstanz, 2008.

[11] Roditty L. On the k shortest simple paths problem in weighted directed graphs [J]. SIAM Journal on Computing, 2010, 39 (6): 2363-2376.

[12] MA Xuan. A genetic algorithm for k optimal paths problem [J]. Computer Engineering and Applications, 2006, 42 (12): 100-103 (in Chinese). [马炫. 求解 K 条最优路径问题的遗传算法 [J]. 计算机工程与应用，2006，42 (12)：100-103.]

[13] HUANG Zehan, TAN Yuejin, DENG Hongzhong. Time-expand network k shortest paths algorithm for over-loading transportations [J]. Computer Engineering and Applications, 2008, 44 (25): 20-23 (in Chinese). [黄泽汉，谭跃进，邓宏钟. 大规模定量传输的时间扩展网络 K 最短路径算法 [J]. 计算机工程与应用，2008，44 (25)：20-23.]

（下转第 3911 页）

术及应用分析 [J]. 技术与市场，2012，12 (4)：92-93.]

[3] TONG Yanqiu, SONG Yang. Design and achieve about wireless online courseware system based on J2ME [J]. Educational Communications and Technology, 2010, 10 (3): 43-49 (in Chinese). [佟延秋，宋阳. 基于J2ME技术的无线网上课件系统的设计与实现 [J]. 教育传播与技术，2010，10 (3)：43-49.]

[4] JING Xin, LU Yao. Design and achieve about dining carte system based on Android [J]. Electronic Science & Technology Review, 2012, 12 (5): 33-34 (in Chinese). [景鑫，陆瑶. 基于Android的餐饮点单系统的设计与实现 [J]. 电子商务，2012，12 (5)：33-34.]

[5] SONG Yongsheng. Design and implementation of the tourist guide system based on Android [J]. Information Technology, 2012, 12 (4): 107-112 (in Chinese). [宋永生. 基于Android的导游系统的设计与实现 [J]. 信息技术，2012，12 (4)：107-112.]

[6] HUANG Jinchuan, JIN Weidong. Study about Web services based on Android [J]. Railway Computer Application, 2010, 19 (11): 24-27 (in Chinese). [黄锦川，金炜东. 基于Android平台Web服务的应用研究 [J]. 铁路计算机应用，2010，19 (11)：24-27.]

[7] YU Zhilong, CHEN Xiaofeng. Google android development getting started and combat [M]. Beijing: Posts&Telecom Press, 2009 (in Chinese). [余志龙，陈小凤. Google android开发入门与实战 [M]. 北京：人民邮电出版社，2009.]

[8] LE Yan, YAO Shanglang, CHEN Xiaofeng, et al. Google android SDK development examples [M]. Beijing: Posts&Telecom Press, 2009 (in Chinese). [勒岩，姚尚郎，陈小凤，等. Google android SDK开发范例大全 [M]. 北京：人民邮电出版社，2009.]

[9] JIANG Yunchen. Principles of android system and practical application [M]. Beijing: Beijing Institute of Technology Press, 2011 (in Chinese). [蒋耘晨. Android系统原理和实战应用 [M]. 北京：北京理工大学出版社，2011.]

[10] ZHU Guiying. Android development applications from the entry to the master [M]. Beijing: Railroad Publishing Press, 2011 (in Chinese). [朱桂英. Android开发应用从入门到精通 [M]. 北京：中国铁道出版社，2011.]

(上接第3906页)

[14] MA Xuan, LIU Qing. A shuffled forg leaping algorithm for k-shortest paths problem [J]. Information and Control, 2011, 40 (5): 614-618 (in Chinese). [马炫，刘庆. 求解k条最短路径问题的混合蛙跳算法 [J]. 信息与控制，2011，40 (5)：614-618.]

[15] LIN Jie, YANG Licai, WU Xiaoqing, et al. Artificial immune optimization method for solving the K-shortest paths search in dynamic route guidance system [J]. Journal of Shandong University, 2007, 37 (2): 103-107 (in Chinese). [林洁，杨立才，吴晓晴，等. 求解动态路径诱导K路最短问题的人工免疫优化方法 [J]. 山东大学学报，2007，37 (2)：103-107.]

K最短路径算法综述

作者： 徐涛， 丁晓璐， 李建伏， XU Tao， DING Xiao-lu， LI Jian-fu

作者单位： 徐涛,丁晓璐,XU Tao,DING Xiao-lu(中国民航大学计算机科学与技术学院,天津300300;中国民航信息技术科研基地,天津300300)， 李建伏,LI Jian-fu(中国民航大学计算机科学与技术学院,天津,300300)

刊名： 计算机工程与设计 PKU

英文刊名： Computer Engineering and Design

年，卷(期)： 2013,34(11)

参考文献(15条)

1. 高松,陆锋 K则最短路径算法效率与精度评估[期刊论文]-中国图象图形学报A 2009(8)

2. Androutsopoulos KN;Zografos KG Solving the k-shortest path problem with time windows in a time varying network[外文期刊] 2008(6)

3. 王增平,李刚,任建文 基于前K最短路径的输电断面搜索新算法[期刊论文]-电工技术学报 2012(4)

4. Hershberger J;Maxel M;Sur S Finding the K shortest simple paths:A new algorithm and its implementation 2007(04)

5. 柴登峰,张登荣 前N条最短路径问题的算法及应用[期刊论文]-浙江大学学报(工学版) 2002(5)

6. 袁红涛,朱美正 K优路径的一种求解算法与实现[期刊论文]-计算机工程与应用 2004(6)

7. 高松,陆锋,段滢滢 一种基于双向搜索的K则最优路径算法[期刊论文]-武汉大学学报(信息科学版) 2008(4)

8. 白轶多,胡鹏,夏兰芳,郭峰林 关于k次短路径问题的分析与求解[期刊论文]-武汉大学学报（信息科学版） 2009(4)

9. Paluch S A multilable algorithm for k shortest paths problem 2009(03)

10. Aljazzar H;Leue S.K * A directed on-the-fly algorithm foe finding the k shortest paths[Technical Report soft-08-03] 2008

11. Roditty L On the k shortest simple paths problem in weighted directed graphs 2010(06)

12. 马炫 求解k条最优路径问题的遗传算法[期刊论文]-计算机工程与应用 2006(12)

13. 黄泽汉,谭跃进,邓宏钟 大规模定量传输的时间扩展网络K最短路径算法[期刊论文]-计算机工程与应用 2008(25)

14. 马炫,刘庆 求解k条最短路径问题的混合蛙跳算法[期刊论文]-信息与控制 2011(5)

15. 林洁,杨立才,吴晓晴,叶杨 求解动态路径诱导K路最短问题的人工免疫优化方法[期刊论文]-山东大学学报（工学版） 2007(2)

引证文献(3条)

1. 面向城市交通网络的K最短路径集合算法[期刊论文]-交通运输系统工程与信息 2014(03)

2. 谢继文,徐涛,姜锡珂,李建伏 基于扩展KMCSP的国际航线运价搜索模型及算法[期刊论文]-计算机工程与设计 2015(08)

3. 付媛,朱礼军,韩红旗 K最短路径算法与应用分析[期刊论文]-情报工程 2015(01)