

文章编号: 1005-8451 (2009) 02-0055-02

大型交通网络最优路径的随机深度搜索算法

张超¹, 刘蕊洁²

(1. 太原铁路局 太原车务段, 太原 030001; 2. 兰州交通大学 数理与软件工程学院, 兰州 730070)

摘要: 提出了随机子空间的概念, 并应用于随机深度搜索算法。该算法可以解决大型交通网络最优路径的搜索问题。并运用于约4000个车站, 520多条线路的交通网络并取得了成效。随机深度搜索算法可以在2000ms快速的找到一个优秀解。

关键词: 随机搜索; 深度搜索; 最短路

中图分类号: U113

文献标识码: A

Random Depth-first Search Algorithm used in optimal path problem of large transport network

ZHANG Chao¹, LIU Rui-jie²

(1. Railway Train Operation Section of Taiyuan Railway Bureau, Taiyuan 030001, China;

2. School of Mathematics, physics & software Engineering Lanzhou JiaoTong University, Lanzhou 730070, China)

Abstract: Wused random depth-first search algorithm to solve optimal path problem of large-scale transport network. We applied the algorithm to the transport network with more than 4,000 stations and 520 lines ,which have achieved success. Random depth-first search algorithm can find an excellent solution in 2000ms.

Key words: random search; depth-first search; optimal path

最短路径问题是图论研究中的一个经典算法问题, 旨在寻找图中任意两结点之间的最短路径。大型交通网络中最优路径的搜索是一个值得解决并深究的问题。当数据量很大、站点关系很复杂时, 传统的搜索方法会受到计算机内存、计算机速度等方面的限制。典型的启发式搜索算法A*算法, 由于并不知道站点之间的相对物理位置, 而无法使用。传统的Dijkstra搜索算法, 深度搜索算法, 广度搜索算法由于数据量巨大导致计算机内存溢出无法正常使用。本文提出的随机深度搜索算法可以有效的解决这一问题。

1 随机深度搜索算法

在搜索空间太大的情况下直接使用传统的宽度优先搜索或深度优先搜索会导致计算机内存溢出, 为此, 本文运用了线路之间关联关系压缩了网路数据, 提出了使用随机子空间进行局部空间搜索的概念。用户能够通过选择不同的权重对换乘车次数、乘车时间和乘车费用等进行评判, 以极快

收稿日期: 2008-09-07

作者简介: 张超, 助理工程师; 刘蕊洁, 在读硕士研究生。

地速度找到优秀的解决方案。

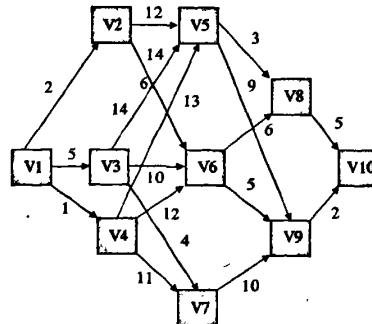


图1 最短路问题

1.1 网路数据的压缩

传统的搜索算法以站点为搜索对象, 当搜索空间巨大, 站点太多, 很容易导致计算机内存溢出。为此, 我们不以站点为搜索对象, 而改用线路作为搜索对象, 从而将搜索空间从4 000多个点压缩到500多个点。我们提出了交通线路之间关联关系的快速算法, 即通过由各个线路所给定的站点所构成的集合作为线路的特征, 通过判断集合的交集维数快速实现线路的关联计算。

1.2 随机子空间局部搜索

为了能够适应更大规模的交通网路系统的查询，本文提出了使用随机子空间进行局部空间搜索的概念。随机子空间，就是每次不是对整个空间进行搜索，而是随机的对其中的一部分进行最优路径的搜索，通过多次搜索基于概率现象，覆盖整个搜索空间，实现对海量数据的搜索。从理论上讲，当搜索次数足够多，根据概率的性质，可以完成对整个空间的搜索。

1.3 随机深度搜索算法

1.3.1 具体实现

我们定义通过始发站的所有线路的集合为始发集合，通过终点站的所有线路的集合为目标集合。从整个（线路）搜索空间随机抽取一个子空间。这个子空间包含目标集合，同时包含若干不隶属于目标集合和始发集合的线路。从始发集合任意选择一条线路。搜索的目的是从这个线路开始，找出互相连接的线路，直至到达目标集合。

判断两条线路是否相交的方法：设 a 线路上所有车站的集合为 A, b 线路上所有车站的集合为 B, 那么这两条线路互相相交的条件是 $\dim(A) + \dim(B) - \dim(A \cup B) > 0$ 其中 $\dim(\cdot)$ 表示集合的维数。

在这个子空间内部随机深度优先搜索，直到获得一个结果。

所谓随机深度优先搜索，指的是在扩展任意一个节点后，我们将扩展的节点以随机顺序，而不是按自然顺序压入待搜索堆栈。每个节点可扩展的条件是：在搜索子空间的尚未被访问的节点（公交线路）中存在与该节点相交的节点。

在找到的一条路里我们用搜索结果是从始发站到终点站的公交线路排列。这个结果以多条互相连接的公交线路组成。由于线路之间往往有多个交点，所以实际上是一个线路束。比如，当始发站为 S, 而终点站为 E, 并且搜索结果是由 L_j 换 L_k 再换 L_m 时，如果 L_j 与 L_k 的交点的集合为 {s₁₁, s₁₂, s₁₃, s₁₄}，而 L_k 与 L_m 的交点的集合为 {s₂₁, s₂₂, s₂₃, s₂₄, s₂₅, s₂₆}，那么，我们的搜索结果可以表示为

S->L_j->{s₁₁, s₁₂, s₁₃, s₁₄}->L_k->{s₂₁, s₂₂, s₂₃, s₂₄, s₂₅, s₂₆}->L_m->E。

这个结果实际上包含 $4^*6=24$ 种走法（线路）。利用动态规划查找搜索结果（线路束）内的最优先

路。多次利用上述过程，力求获得优秀解。

1.3.2 实验分析

我们以约 4 000 个车站，520 多条线路的交通网络为实验对象。所有的车站的站名及线路名称符号化，隐瞒了具体车站的物理位置。520 条线路中包括上行线，下行线，逆行线和环行线四种线路，票价分为单一票价与分段计价两种。运用 java 语言，应用随机深度搜索算法得到任意起始站—终到站之间的乘车时间最短、换乘次数最少、乘车费用最小的路线。以下以 S3359 → S1828 为例，图二为程序运行结果。



图 2 运行结果

通过图 2，在搜索耗时上，随机深度搜索算法可以在 2 000 ms 左右得到搜索路径的优秀解。

2 结束语

本文提出了随机深度搜索，解决了大型交通网络的最优选路问题。通过在约 4 000 个车站，500 条线路的交通网络上实验分析，随机深度搜索可以快速的找到一个优秀解。随机搜索算法当搜索次数足够多的时候是可以求得最优解的，但当搜索次数减少的时候怎样求得最优解还有待进一步研究。

参考文献：

- [1] 徐士良. 计算机常用算法[M]. 北京：清华大学出版社，1995.
- [2] 绝培明. 矢量网络模型中最短距离算法的研究[EB/OL]. 中文核心期刊网.
- [3] 周琳. 数字化交通地图中最短路径算法的改进[J]. 通讯与计算机, 2006.