

改进的 Dijkstra 最短路径算法及其应用研究

王树西 吴政学

(对外经济贸易大学信息学院 北京 100029)

摘 要 求最短路径是一个应用很广泛的问题。求最短路径的算法有很多,公认较好的算法是 Dijkstra 标号法。但实验结果表明,Dijkstra 标号法有需要改进的地方:①其退出机制对不联通的有向图是无效的,会陷入死循环;②没有涉及最短路径上顶点的邻接点(特指前面的相邻点)问题;③没有涉及多个顶点同时获得 p 标号的问题。针对上述问题,对标号法进行了改进。算法实验表明,改进的标号法能够有效解决上述问题。在上述工作的基础上,开发了“北京市道路最优路线选择系统”,以提供起点和终点之间的最优路线,帮助用户选择出行路线,使市民能够避开交通最拥堵的路段,节约出行时间。

关键词 最短路径,Dijkstra 标号法,城市交通,最优路线选择

中图法分类号 TP392 文献标识码 A

Improved Dijkstra Shortest Path Algorithm and its Application

WANG Shu-xi WU Zheng-xue

(Academy of Information, University of International Business and Economics, Beijing 100029, China)

Abstract The problem of "the shortest path" has wide application. There are many algorithms to solve this problem and the best one is "Dijkstra's label algorithm". But experiment results show that this algorithm needs to be improved: ①The algorithm's exit mechanism is ineffective to non-unicom figure and will fall into an infinite loop; ②The algorithm is not concerned in the problem of previous adjacent vertices in shortest path; ③The algorithm is not concerned in the problem that more vertices obtain the "p-label" at the same time. This paper improved the "Dijkstra's label algorithm". Experiment results indicate that the improved algorithm can solve above problems effectively. On the basis of the above work, we developed the "Beijing Road optimal route selection system", which help users to select the shorted route, so that people can avoid the most congested road traffic and save time.

Keywords Shortest path, Dijkstra's labeled algorithm, Urban transport, Optimal route selection

1 引言

在现实生活、生产、组织、管理工作中,需要解决许多优化(最短路径)问题。例如在企业管理中,制定合理的管理计划和设备购置计划,使收益大而费用小;在组织生产中,将各工序衔接好,使生产任务完成得既快又好;在现有交通网络中,安排合理的交通路径,使得调运的物资数量多而费用小。这些问题,都可以归结为最短路径问题。

求解给定网络图中某两点(一般常指始点和终点)之间的最短路径问题,广泛应用于各个领域中,例如计算机网络路由算法、机器人探路、交通路线导航、游戏设计等。

本文首先介绍求最短路径的国内外研究现状,然后简要介绍了 Dijkstra 标号法,并通过实验指出 Dijkstra 标号法需要改进的地方。在此基础上,提出了改进的算法,并通过实验对改进的算法进行了验证。实验结果表明,改进之后的算法可以有效解决标号法的不足。在以上工作的基础上,开发了“北京市道路最优路线选择系统”,以提供起点和终点之间的

最优路线,帮助用户选择出行路线,使市民能够避开交通最拥堵的路段,节约出行时间。系统测试结果表明,改进后的 Dijkstra 标号法可以得到有效应用。

2 最短路径算法的国内外研究现状

最短路径问题,是图论研究中的一个经典算法问题,它旨在寻找图(由结点和路径组成的)中两结点之间的最短路径。也就是说,从图中某一顶点(源点)到达另一顶点(终点)的路径可能不止一条,如何找到一条路径使得沿此路径上各边的权值总和(称为路径长度)达到最小。

最短路径算法具体的形式包括:

①确定起点的最短路径问题。即已知起始结点,求最短路径。本问题可以形式化为已知图 $G=(V, E)$, 找出从某给定的源结点 $S \in V$ 到 V 中的每个结点的最短路径。

②确定终点的最短路径问题。与确定起点的问题相反,该问题是已知终结结点,求最短路径。在无向图中,该问题与确定起点的问题完全等同;在有向图中,该问题等同于把所有

路径方向反转的确定起点的问题。

③确定起点、终点的最短路径问题。即已知起点和终点，求两结点之间的最短路径。

④全局最短路径问题，即求图中所有的最短路径。
用于解决最短路径问题的算法被称作“最短路径算法”。最短路径算法很多，常用的路径算法有①Dijkstra 算法；②A* 算法；③Bellman-Ford 算法；④SPFA 算法（Bellman-Ford 算法的改进版本）；⑤Floyd-Warshall 算法；⑥Johnson 算法；⑦Bi-Direction BFS 算法。

上述求解最短路径的算法，大致可以分为：①静态路径最短路径算法；②动态路径最短路径算法。静态路径最短路径算法，是指外界环境不变，计算最短路径，主要有 Dijkstra 的标号法^[1]、A*（A Star）算法等。动态路径最短路径算法，是指外界环境不断发生变化，在不能预测的情况下计算最短路径，如游戏中敌人或障碍物不断移动的情况下求最短路径，典型的有 D* 算法。美国火星探测器核心的寻路算法就是采用的 D*（D Star）算法。本文只讨论静态路径最短路径算法。

目前国内外对最短路径问题的研究有很多，目前的研究倾向于应用研究。文献[2]提出一种高效的最短路径树动态更新算法；文献[3]提出一种新的道路网络连续查询处理方法；文献[4]对可伸缩的增量连续 k 近邻查询处理进行了研究；文献[5]对基于数据网格的书法字 k 近邻查询进行了研究；文献[6]研究了空间网络数据库中最近邻查询的设计与实现；文献[7]对基于道路网络的对象聚类进行了研究；文献[8]对动态网络最短路径问题的复杂性及近似算法进行了研究；文献[9]研究了卫星时变拓扑网络最短路径算法；文献[10]提出了一种基于改进最小生成树算法的配电网架优化规划；文献[11]对时空数据库中基于 TPR-树的反向最近邻查询进行了研究；文献[12]研究了道路网络中的连续最近邻查询。

3 Dijkstra 的标号法

到目前为止，公认的求最短路径的较好算法是由 Dijkstra 于 1959 年给出的标号法。

标号法应用的范围非常广泛，例如在多点路由上的应用，在测绘科学中的应用，在物流运输最短路径中的应用，在智能交通系统中的应用，在高速公路联网收费中的应用等。

针对 Dijkstra 标号法的改进算法、求最短路径的算法，也有很多相关研究。

3.1 Dijkstra 标号法的算法内容^[1]

设 $G=\langle V,E,W\rangle$ 是 n 阶简单带权图， $w_{ij}\geq 0$ 。如果顶点 v_i 与 v_j 不相邻，令 $w_{ij}=\infty$ ，求 G 中顶点 v_1 到其余各顶点的最短路径（其实可以求任意两顶点之间的最短路径），为此，先给出下面的定义。

- (1) 设 $l_i^{(r)*}$ 为顶点 v_1 到顶点 v_i 最短路径的权，若顶点 v_i 获得了标号 $l_i^{(r)*}$ ，则称 v_i 在第 r 步获得了 p 标号 $l_i^{(r)*}$ （永久性标号），其中 $r\geq 0$ 。
- (2) 设 $l_j^{(r)}$ 为 v_1 到顶点 v_j 的最短路径权的上界，如果 v_j 获得 $l_j^{(r)}$ ，则称 v_j 在第 r 步获得 t 标号 $l_j^{(r)}$ （临时性标号）。
- (3) 设 $P_r=\{v|v \text{ 已获得 } p \text{ 标号}\}$ 为第 r 步通过集。
- (4) 设 $T_r=V-P_r$ 为第 r 步未通过集， $r\geq 0$ 。

Dijkstra(标号法)的算法如下：

开始， $r\leftarrow 0$ ， v_1 获 p 标号： $l_1^{(0)*}=0$ ， $P_0=\{v_1\}$ ， $T_0=V-\{v_1\}$ ， v_j ($j\neq 1$)

的 t 标号： $l_j^{(0)}=w_{1j}$ 。

①求下一个 p 标号顶点。

设 $l_i^{(r)*}=\min_{v_j\in T_{r-1}}\{l_j^{(r-1)}\}$ ， $r\geq 1$ 。将 $l_i^{(r)*}$ 标在相应顶点 v_i 处，表明 v_i 获得 p 标号，修改通过集和未通过集： $P_r=P_{r-1}\cup\{v_i\}$ ， $T_r=T_{r-1}-\{v_i\}$ 。

查 T_r ：若 $T_r=\emptyset$ ，则算法结束，否则转②。

②修改 T_r 中各顶点的 t 标号：

$l_j^{(r)}=\min\{l_j^{(r-1)}, l_i^{(r)*}+w_{ij}\}$ ， $l_i^{(r)*}$ 是刚刚获得 p 标号顶点的 p 标号。令 $r\leftarrow r+1$ ，转①。

3.2 无向联通带权图的算法实验

Dijkstra 的标号法能够有效解决无向联通带权图的最短路径问题。图 1 是一个无向带权联通图，根据 Dijkstra 的标号法，求 v_0 与 v_5 的最短路径。

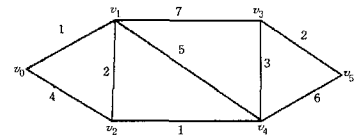


图 1

用标号法求 v_0 与其余各点的最短路径。计算过程见表 1。

表 1

$v_i \backslash r$	v_0	v_1	v_2	v_3	v_4	v_5
0	0	1	4	∞	∞	∞
1		$\boxed{1}/v_0$	3	8	6	∞
2			$\boxed{3}/v_1$	8	4	∞
3				7	$\boxed{4}/v_2$	10
4				$\boxed{7}/v_4$		9
5						$\boxed{9}/v_3$
W	0	1	3	7	4	9

至此，全部顶点都已得到标号，计算结束。得到点 v_0 到 v_1, v_2, v_3, v_4, v_5 的最短路径分别为 $v_0 v_1, v_0 v_1 v_2, v_0 v_1 v_2 v_4 v_3, v_0 v_1 v_2 v_4, v_0 v_1 v_2 v_4 v_3 v_5$ ，其长度分别为 1、3、7、4、9。

4 Dijkstra 标号法需要改进之处

Dijkstra 标号法有一些需要改进的地方，下面逐一说明，并提出相应的改进方法。

4.1 退出机制

在 Dijkstra 标号法中，算法的退出机制是：查第 r 步未通过集 T_r ($r\geq 0$)，若 $T_r=\emptyset$ ，则算法结束。

这种退出机制对于无向带权联通图是有效的，但是在有向带权图中，如果两个点之间是不连通的，那么根据现有算法的退出机制，系统将无法停机而陷入死循环。

例如，在图 2(有向带权图)中，如果根据 Dijkstra 标号法求 v_0 与 v_4 的最短路径，系统就会无法停机而陷入死循环，因为 v_0 与 v_4 这两点之间是不连通的。也就是说，标号法对有向图可能无效，甚至会陷入死循环。

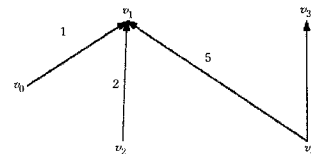


图 2

针对这一点，本文提出如下退出机制：查第 r 步未通过集

$T_r(r \geq 0)$, 若 $T_r = \emptyset$, 则算法结束; 求 p 标号顶点, 设 $l_i^{(r)*} = \min_{v_j \in T_{r-1}} \{l_j^{(r-1)}\} (r \geq 1)$, 如果 $l_i^{(r)*} = \infty$, 那么算法结束。

4.2 最短路经上顶点的相邻点

在 Dijkstra 标号法中, 并没有明确指出如何求取最短路经上顶点的相邻点(特指前面的相邻点), 而在实际应用中, 往往要求出最短路经上顶点的相邻点。所以在这方面, Dijkstra 标号法需要改进。

本文提出如下改进措施: 根据顶点 v_i 修改 T_r 中各顶点(不妨设为 v_j)的 t 标号时, 如果点 v_j 的 t 标号被更新, 那么应该标明 v_j 前面的相邻点是 v_i 。每个点 v_j 都可能有多多个前面的相邻点。

4.3 多个顶点可以同时获得 p 标号

Dijkstra 的标号法忽略了一个问题, 即多个顶点可以同时获得 p 标号。所以在这方面, Dijkstra 标号法需要改进。

本文提出如下改进措施: 设 $l_i^{(r)*} = \min_{v_j \in T_{r-1}} \{l_j^{(r-1)}\} (r \geq 1)$ 。将 $l_i^{(r)*}$ 标在相应顶点 v_i 处, 表明 v_i 获得 p 标号, 修改通过集和未通过集: $P_r = P_{r-1} \cup \{v_i\}$, $T_r = T_{r-1} - \{v_i\}$ 。如果多个点的 t 标号相同, 那么所有这些点同时获得 p 标号。

5 改进的 Dijkstra 标号法

根据 Dijkstra 需要改进的地方, 以及相应的改进措施, 本文提出一种改进的 Dijkstra 标号法。

5.1 基本定义

(1) 设 $l_i^{(r)*}$ 为顶点 v_1 到顶点 v_i 最短路经的权, 若顶点 v_i 获得了标号 $l_i^{(r)*}$, 则称 v_i 在第 r 步获得了 p 标号 $l_i^{(r)*}$ (永久性标号), 其中 $r \geq 0$ 。

(2) 设 $l_j^{(r)}$ 为顶点 v_1 到顶点 v_j 的最短路经权的上界, 如果 v_j 获得 $l_j^{(r)}$, 则称 v_j 在第 r 步获得 t 标号 $l_j^{(r)}$ (临时性标号)。

(3) 设 $P_r = \{v | v \text{ 已获得 } p \text{ 标号}\}$ 为第 r 步通过集。

(4) 设 $T_r = V - P_r$ 为第 r 步未通过集, $r \geq 0$ 。

(5) 设 A_i 为顶点 v_i 前面的邻接点集合。

(6) 设 N_r 为第 r 步获得 p 标号的点集合。

5.2 具体的改进算法

初始化: $r \leftarrow 0$, v_1 获 p 标号: $l_1^{(0)*} = 0$, $P_0 = \{v_1\}$, $T_0 = V - \{v_1\}$, $v_j (j \neq 1)$ 的 t 标号: $l_j^{(0)} = w_{1j}$, 如果 $l_j^{(0)} \neq \infty$, 那么 $A_j = \{v_1\}$, 否则 $A_j = \emptyset$ 。

① 求下一个 p 标号顶点。

设 $\min l^{(r-1)} = \min_{v_j \in T_{r-1}} \{l_j^{(r-1)}\}$, $r \geq 1$, $N_r = \emptyset$ 。

如果 $\min l^{(r-1)} = \infty$, 则算法结束。

查 $v_i \in T_{r-1}$: 如果 $l_i^{(r-1)} = \min l^{(r-1)}$, 那么 v_i 获得 p 标号: $l_i^{(r)*} = \min l^{(r-1)}$ 。

修改: 通过集 $P_r = P_{r-1} \cup \{v_i\}$, 未通过集 $T_r = T_{r-1} - \{v_i\}$, $N_r = N_r \cup \{v_i\}$ 。

查 T_r : 若 $T_r = \emptyset$, 则算法结束, 否则转②。

② 根据 N_r , 修改 T_r 中各顶点的 t 标号:

对于 $v_j \in T_r$, $l_j^{(r)} = l_j^{(r-1)}$,

对于 $v_i \in N_r$,

如 $(l_i^{(r)*} + w_{ij}) < l_j^{(r)}$, 则 $l_j^{(r)} = (l_i^{(r)*} + w_{ij})$, $A_j = \{v_i\}$

如果 $(l_i^{(r)*} + w_{ij}) = l_j^{(r)}$, 那么 $A_j = A_j \cup \{v_i\}$

令 $r \leftarrow r + 1$, 转①。

6 对改进之后的算法进行实验

根据改进之后的 Dijkstra 标号法, 本文进行了一系列算法实验。

6.1 实验 1: 求顶点 v_1 到其余各点的最短路经(见图 3)

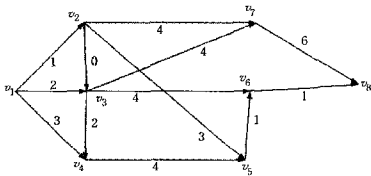


图 3

用改进之后的标号法进行实验。实验过程见表 2。

表 2

v_i	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
1	$\boxed{0}/v_1$	$1/v_1$	$2/v_1$	$3/v_1$	∞	∞	∞	∞
2		$\boxed{1}/v_1$	$1/v_2$	$3/v_1$	$4/v_2$	∞	$5/v_2$	∞
3			$\boxed{2}/v_2$	$3/v_1, v_3$	$4/v_2$	$5/v_3$	$5/v_2, v_3$	∞
4				$\boxed{3}/v_1, v_3$	$4/v_2$	$5/v_3$	$5/v_2, v_3$	∞
5					$\boxed{4}/v_2$	$5/v_3, v_5$	$5/v_2, v_3$	∞
6						$\boxed{5}/v_3, v_5$	$\boxed{6}/v_2, v_3$	$6/v_6$
7							$\boxed{7}/v_6$	

实验分析: 顶点可能有多多个邻接点, 例如第 3 步中, v_4 有 2 个邻接点: v_1, v_3 。相应地, 最短路经可能有多条。

6.2 实验 2: 求顶点 v_1 到其余各点的最短路经(见图 4)

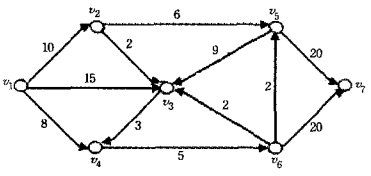


图 4

用改进之后的标号法进行实验。实验过程见表 3。

表 3

v_i	v_1	v_2	v_3	v_4	v_5	v_6	v_7
1	$\boxed{0}/v_1$	$10/v_1$	$15/v_1$	$8/v_1$	∞	∞	∞
2		$\boxed{10}/v_1$	$15/v_1$	$\boxed{8}/v_1$	∞	$13/v_4$	∞
3			$\boxed{12}/v_1$	$12/v_2$	$16/v_2$	$13/v_4$	∞
4				$\boxed{8}/v_2$	$16/v_2$	$13/v_4$	∞
5					$\boxed{15}/v_6$	$\boxed{13}/v_4$	$33/v_6$
6						$\boxed{16}/v_6$	$33/v_6$
7							$\boxed{33}/v_6$

实验分析: 在本实验中, 图形是有向图。

6.3 实验 3: 求顶点 b 到其余各点的最短路经(见图 5)

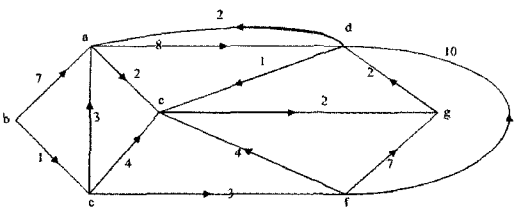


图 5

用改进之后的标号法进行实验。实验过程见表 4。

实验分析: 在本实验中, 多个顶点可以同时获得 p 标号。在第 3 步中, a, f 两个点同时获得 p 标号。

表 4

$r \backslash v_i$	b	a	c	d	e	f	g
1	∞	7/b	1/b	∞	∞	∞	∞
2		4/c	∞ /b	∞	5/c	4/c	∞
3		∞ /c		12/a	5/c	∞ /c	11/f
4				12/a	∞ /c		7/e
5				9/g			∞ /e
6				∞ /g			

7 算法应用

最短路径算法的一个重要应用是交通网络。交通网络可以画成带权的图,图中顶点表示城市,边代表城市间的公路,边上的权表示公路的长度。

对于这样的交通网络常常提出这样的问题:两地之间是否有公路可通?在有几条路可通的情况下,哪一条路最短?以上提出的问题就是在带权图中求最短路径问题,此时路径的长度不是路径上边的数目,而是路径上的边所带权值的总和。

随着城市化进展速度的加快,市民面临的交通压力也越来越大,如何为市民提供更加方便快捷的路线服务,使市民能够避开交通最拥堵的路段,节约出行时间,已经成为城市交通管理的重要课题。

本文根据改进的 Dijkstra 标号法,以北京市为例,开发了“北京市道路最优路线选择系统”,以提供起点和终点之间的最优路线,帮助用户选择出行路线,使市民能够避开交通最拥堵的路段,节约出行时间。

7.1 系统需求分析

城市交通是城市的血脉,如果城市血脉拥堵,城市就会瘫痪。目前,城市交通拥堵已严重影响人们的日常生活,这不仅表现在大城市,二、三线城市也凸现拥堵现象。为了缓解交通压力,方便人们出行,相关部门采取了一系列措施,如运用信息系统实现对交通的信息化管理,通过电台、电视、网络等渠道实时发布各路段交通状况。这种方式虽然能让用户了解到最新的路段信息,但缺乏对用户需求的深入研究,而且通过各媒体发布的信息太零散,用户很难根据了解到的路段信息选择出行路线,即使是熟悉城市交通路况的有经验的司机,也很难选择出路程短且流畅的最优路线。

“北京市道路最优路线选择系统”,是基于北京市公安局公安交通管理局网站系统建立的,可使用户能够方便、快捷地根据自己的偏好选择最优路线。路线系统整合了当前时段的各路段拥堵情况,大大提高了用户进行路线选择的效率和质量,完整准确的信息和方便快捷的操作流程,为用户节约了时间;信息管理技术在城市交通领域的运用具有重要的理论意义和实践意义。

系统目标是:1)通过为用户提供简单、快捷、详细、准确的路线选择服务,使用户可以根据当前的交通状况结合自身偏好选择最优路线;2)实现道路信息管理的规范化,便于数据分析和统计分析;3)充分发挥信息管理系统的优势,减轻人工负担,提高效率。

7.2 系统简介

“北京市道路最优路线选择系统”,是使用 JSP 技术和 MySQL 数据库作为开发工具的基于 B/S 架构开发的网站。

系统截图如图 6 所示。

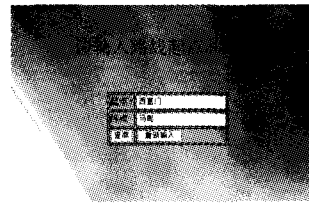


图 6

本系统将交管局网站上所有的路段信息集成到 MySQL 数据库中,包括路线、关闭路段、交通流量等,然后运用 JSP 语言调用数据库中的信息,进行数据库的基本的增、删、改、查操作。

本系统将所有路线简化为多个节点组成的路线图,每个节点对应于一个地点。在进行路径选择时,遍历节点路线图,调用数据库当前信息,统计出每条路线的当前数据,包括交通流量、长度等,然后输出最优路线。

系统管理员通过操作数据库,可以完成对各路段查询次数、各时间段交通流量等信息进行统计,为相关政策制定、研究提供数据参考。

市民运用系统可以根据最新路况进行路线选择,节约时间成本,合理安排出行,不仅提高了自身的出行效率,也有利于减少拥堵路段的交通流量,对整个城市交通的通畅有着重要的意义。

系统的主要板块有交管信息、实时路况、网上办事、为您指路、网上咨询、交通安全、交警风采。

用户访问本网站,可以查看北京市公安局、公安交通管理局、交管局介绍、动态信息、警务公开、交通法规、交通通告及进行车辆违章查询及北京驾驶员积分查询等操作。

7.3 系统业务流程图

1) 用户业务流程图(见图 7)

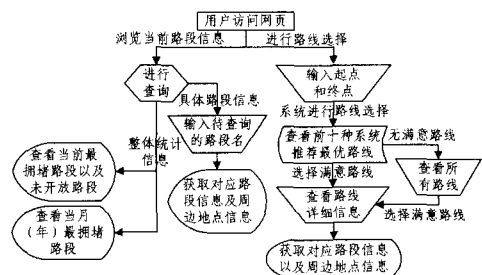


图 7

2) 管理员业务流程图(见图 8)

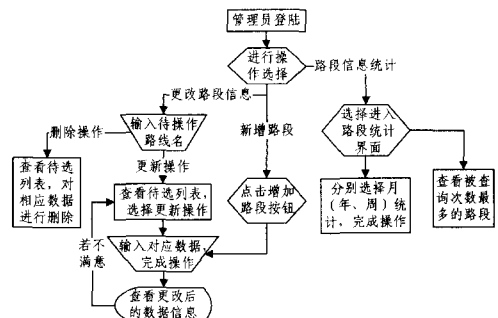


图 8

3) 系统数据流程图(见图 9)

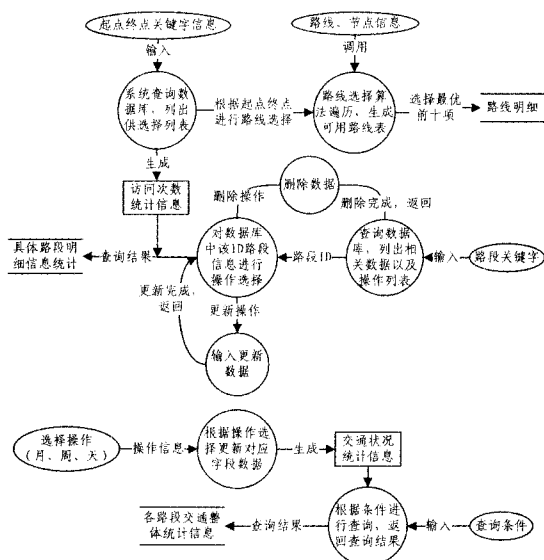


图 9

7.4 系统设计

1)ER 图(见图 10)

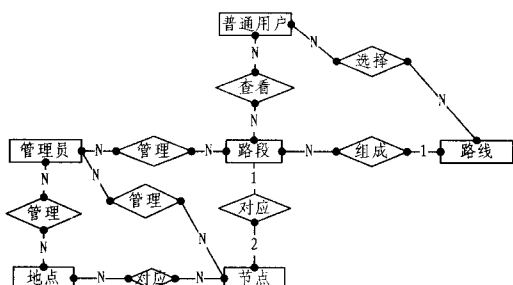


图 10

2)数据库物理模型图(见图 11)

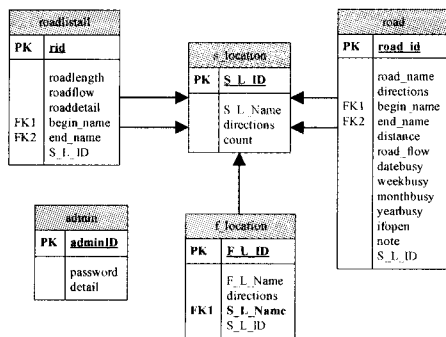


图 11

7.5 系统测试

在“起点”文本框输入“西直门”,在“终点”文本框输入“马甸”之后,系统推荐前 10 条最优路线。输出结果如图 12 所示。

路線編號	路線長度 (公里)	路線維持度	最佳成績	最佳成績月份
105	2.0	7.0	9.4	12月
107	10.7	3.0	13.7	12月
108	10.0	4.0	14.0	12月
110	18.1	5.0	20.1	12月
104	14.1	5.0	20.6	12月
114	18.8	6.0	24.3	12月
83	19.1	5.0	26.1	12月
18	19.8	7.0	26.9	12月
109	20.0	7.0	27.0	12月
103	20.6	7.0	27.5	12月

資料來源：根據1980年度改善計畫所記錄

图 12

用户可以选择路线,查看详细信息,还可以按照自己的偏好(距离最短或者交通最流畅)查询符合条件的路线。若系统推荐的前 10 条路线中没有满意的路线,则可以点击查看所有路线,如图 13 所示。

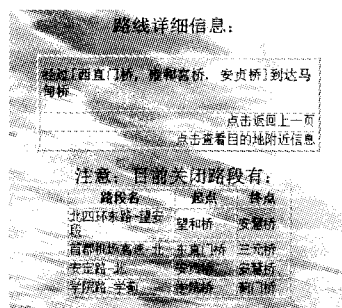


图 13

详细信息中包括目前已关闭路段(因交通管制或者交通事故),其可作为用户路线选择的参考。

7.6 系统小结

本系统实现了两地之间最优路线的选择,用户可以根据自己的偏好选择最短路线或者最快路线到达目的地,基本满足了普通市民出行的需求,同时完成了对各路段交通流量信息的系统化管理,方便了交通部门进行查询、统计。

本系统的核心在于根据实时交通信息进行的带权路径选择,算法每次从数据库调用最新数据进行运算,将所有可能路线写入数据库临时表中,然后根据用户偏好加入权值,列出最优路线。在系统实现过程中,为了避免由于算法时间复杂度偏大导致速度减慢,在数据库中增加了节点表,其与地点表相对应。将北京交通图简化到 30 多个节点,缩短了运算时间,降低了硬件的成本,同时方便了用户了解目的地周边信息。

结束语 最短路径问题,是图论研究中的一个经典算法问题。Dijkstra 算法,是解决最短路径问题的较好算法。

本文对 Dijkstra 标号法进行了分析,通过实验指出了其中的不足之处,并提出了改进的方法。在此基础上,提出了改进的 Dijkstra 最短路径算法,并进行了一系列针对性的实验。实验结果表明,改进的 Dijkstra 标号法不但可以解决连通无向带权图的最短路径问题,而且可以解决连通有向带权图的最短路径问题。改进之后的 Dijkstra 标号法,在算法退出机制上做了改进,从而使得算法免于陷入死循环;可以有效查询最短路径上顶点的邻接点;解决了多个顶点同时获得 p 标号的问题。

也就是说,该文对 Dijkstra 算法(只适用于无向连通带权图)进行了改进。改进的 Dijkstra 算法既对无向连通带权图适用,也对有向连通带权图适用,对阶数较小的图通过实验证明也是有效的。

在以上工作的基础上,开发了“北京市道路最优路线选择系统”,以提供起点和终点之间的最优路线,帮助用户选择出行路线,使市民能够避开交通最拥堵的路段,节约出行时间。系统测试结果表明,改进后的 Dijkstra 标号法可以得到具体的应用,有一定的应用价值。

但是,本文也存在很多的不足:

1)改进的 Dijkstra 算法,对阶数较大的图是否有效,需要进一步严格的证明。

2)实事求是地讲,Dijkstra 算法的时间复杂度为 $O(n^2m)$,并不是太高。改进的 Dijkstra 算法,在时间复杂度方面,并没有明显改进。

所以,下一步的工作,将着重从如下方面入手:

1)对改进的 Dijkstra 算法进行严格证明,确保对阶数较大的图有效。

2)对 Dijkstra 算法继续改进,改进 Dijkstra 算法的时间复杂度,提高 Dijkstra 的运行效率。这将是具有理论意义的。

3)Dijkstra 算法应用非常广泛。下一步的工作,要对 Dijkstra 标号法的应用进行深入研究。这将是具有使用意义的。

参 考 文 献

[1] Dijkstra E. A note two problems in connection with graphs[J]. Numerical Mathemat,1959,1:269-271
[2] 刘代波,侯孟书,武泽旭,等. 一种高效的最短路径树动态更新算法[J]. 计算机科学,2011,38(7):96-99
[3] 廖巍,吴晓平,严承华,等. 一种新的道路网络连续查询处理方法

[J]. 计算机科学,2009,36(9):151-153
[4] 廖巍,熊伟,王钧,等. 可伸缩的增量连续 k 近邻查询处理[J]. 软件学报,2007,18(2):268-278
[5] 庄毅,庄越挺,吴飞. 基于数据网格的书法字 k 近邻查询[J]. 软件学报,2006,17(12):2289-2301
[6] 孙亚. 空间网络数据库中最近邻查询的设计与实现[J]. 计算机科学,2008,35(3):73-75
[7] 陈继东,孟小峰,赖彩凤. 基于道路网络的对象聚类[J]. 软件学报,2007,18(2):332-344
[8] 林澜,闫春钢,蒋昌俊,等. 动态网络最短路问题的复杂性与近似算法[J]. 计算机学报,2007,30(4):608-614
[9] 张涛,柳重堪,张军. 卫星时变拓扑网络最短路径算法研究[J]. 计算机学报,2006,29(3):371-377
[10] 刘健,杨文宇,余健明,等. 一种基于改进最小生成树算法的配电网架优化规划[J]. 中国电机工程学报,2004,24(10):103-108
[11] 于海彦,郝忠孝. 时空数据库中基于 TPR-树的反向最近邻查询[J]. 哈尔滨理工大学学报,2007,12(3):87-90
[12] 冯惠妍,郭俊凤. 道路网络中的连续最近邻查询[J]. 计算机工程,2010,36(8):79-82

(上接第 200 页)

结果精度低于 10^{-12} ,则认为结果为 0。从表 4 中数据可以看出,PGSO 在 7 个函数中的结果优于其它 3 种算法,在 9 个函数中搜索到了全局最优值。综上所述,经过大量的实验对比分析,结果表明本文提出的改进 GSO 算法是求解高维全局优化问题的有效方法。

表 4 PGSO 与其他算法的比较(D=30)

	GA ^[3]	PSO ^[3]	DE ^[3]	PGSO
Sphere	1.11E+03	0.00E+00	0.00E+00	0.00E+00
Rosenbrock	1.96E+05	1.51E+01	1.82E+01	2.12E-04
Griewank	1.06E+01	1.74E-02	1.48E-03	0.00E+00
Rastrigin	5.29E+01	4.40E+01	1.17E+01	0.00E+00
Schwefel2.26	9.76E+02	5.56E+03	2.30E+03	3.82E-04
Ackley	1.47E+01	1.65E-01	0.00E+00	0.00E+00
Step	1.17E+03	0.00E+00	0.00E+00	0.00E+00
Quartic	1.81E-01	1.16E-03	1.36E-03	9.06E-05
Schwefel2.22	1.10E+01	0.00E+00	0.00E+00	0.00E+00
Schwefel1.2	7.40E+03	0.00E+00	0.00E+00	0.00E+00
Dixon-Price	1.22E+03	6.67E-01	6.67E-01	0.00E+00
Penalized	1.25E+02	7.68E-03	2.20E-03	0.00E+00

结束语 基于群智能理论的优化方法研究方兴未艾。萤火虫群优化算法作为新型的优化方法,在算法和应用方面的研究刚刚起步。本文针对 GSO 算法中存在的收敛精度低的问题,基于群智能理论,借鉴人工蜂群算法和粒子群算法中个体生成机制,对 GSO 算法的移动策略和移动方式进行了改进,提出了更优接受策略和逐维更新机制。通过对标准测试函数进行实验仿真,并与基本 GSO 算法和其他算法比较,结果表明本文提出的改进算法能大幅提高 GSO 算法的收敛精度,是求解高维全局优化问题的有效方法。

参 考 文 献

[1] Krishnanand K N,Ghose D. Detection of multiple source locations using a glowworm metaphor with applications to collective

robotics [C]//IEEE Swarm Intelligence Symposium. Pasadena, California:IEEE Press,2005:84-91
[2] Krishnanand K N,Ghose D. Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations [J]. Robotics and Autonomous Systems,2008,56(7):549-569
[3] Krishnanand K N,Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions[J]. Swarm Intelligence,2009,3(2):87-124
[4] Krishnanand K N,Ghose D. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications[J]. Multi-agent and Grid Systems,2006,2(3):209-222
[5] Kaipa K N,Puttappa A,et al. Rendezvous of glowworm-inspired robot swarms at multiple source locations:A sound source based real-robot implementation[C]//Lecture Notes in Computer Science. 2006,4150:259-269
[6] Bharat T V. Agents based algorithms for design parameter estimation in contaminant transport inverse problems [C]// 2008 IEEE Swarm Intelligence Symposium (SIS 2008). St. Louis, MO, United States,2008:1-7
[7] He D, Zhu H. Glowworm swarm optimization algorithm based on multi-population [C]// Proceedings-2010 6th International Conference on Natural Computation(ICNC 2010). Yantai, Shandong, China, IEEE Computer Society,2010:2624-2627
[8] 李咏梅,周永权,姚祥光. 基于追尾行为的改进型人工萤火虫群算法[J]. 计算机科学,2011,38(3):248-251
[9] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm [J]. Applied Mathematics and Computation, 2009, 214:108-132
[10] 方开泰. 均匀设计与均匀设计表[M]. 北京:科学出版社,1994:74-97