Ford 算法的改进算法

赵礼峰,梁 娟
(南京邮电大学 理学院,江苏 南京 210023)

摘 要:Ford 算法是求解不含负回路网络中从源节点到其余各节点最短路径的经典算法。但每次逼近中,都要计算所有节点的入弧,重复计算量大,降低了计算效率。为此,文中通过引入两个数组和只计算权值变小的节点的所有出弧对 Ford 算法进行改进,改进后的算法既能快速地计算从源节点到其余各节点的最短路权值,又能更直观地找出最短路径。最后通过具体实例分析和仿真结果表明,改进算法不仅简化了计算量,降低了时间复杂度,而且增强了寻路直观性。

关键词:最短路;Ford 算法;不含负回路网络;改进算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2015)07-0072-04
doi:10.3969/j.issn.1673-629X.2015.07.016

Improved Algorithm of Ford Algorithm

ZHAO Li-feng, LIANG Juan
(College of Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: Ford algorithm is a classical algorithm that finds the shortest path from the source node to other nodes in solving the network without negative loop. However, all incoming arcs weights are needed to be calculated from all nodes in each approximation, and the amount of repeated calculation increases which decreases the efficiency of the algorithm. Ford algorithm is improved in this paper by introducing two arrays and calculating all outgoing arcs weights from nodes whose weight become smaller. The improved algorithm can both calculate the shortest path weights more quickly and find the shortest paths more directly from the source node to other nodes. Finally, the specific analysis and simulation results indicate that the improved algorithm not only simplifies the amount of calculation and reduces the time complexity, but also enhances the intuition of finding the shortest path.

Key words: the shortest path; Ford algorithm; network without negative loop; improved algorithm

0 引 言

近年来,随着科学技术和网络技术的发展,人们对生产生活效率的要求逐渐提高,最短路问题在运筹学、信息论、控制论以及计算机科学等各个领域,发挥着越来越重要的作用,国内外很多行业的大量专家都对这一问题做了深入的研究[1-7],并且取得了很大的进展。其中 Dijkstra 算法可以直接用来求解两个指定节点之间或指定节点到其余各节点之间的最短路问题[8-9],但对于含有负权值的网络,Dijkstra 算法是失效的;Ford 算法是求指定节点到其余各节点最短路的一种经典算法[10-12]。对于求解从指定节点到其余各节点最短路径的问题,有两种算法:第一,先用 Dijkstra 算法求解指定节点到另一个节点的最短路径,重复使用 Dijkstra 算法 $n-1$ 次,即可求出指定节点到其余所有节点的最短路径,但比较繁琐;第二,用 Ford 算法直接求解指定节点到其余所有节点的最短路径问题。Ford 算法的思想是逐次逼近,每次逼近都要计算所有节点的入弧的权值[13],其中有些计算是不必要的,因为若本次逼近中与该节点所有入弧相连的节点的权值均不变,则下一次逼近时,计算所得该节点的权值也不会改变,所以下一次逼近中该节点所有入弧的计算量都可以省略。另外,数据的存储方式使得 Ford 算法在应用计算机语言实现的过程中不够优化[14]。

针对上述不足,文中对 Ford 算法做了一些改进。一方面,改进算法在计算最短路时,对 Ford 算法中那些与某个节点 v_i 所有入弧相连的节点的权值均不变的节点,都不参与计算,只计算权值变小的节点 v_k,用该节点的权值 u_k 加上 w_{kj}($0 < w_{kj} < \infty$,即节点 v_k 的所

收稿日期:2014-08-13 **修回日期**:2014-11-20 **网络出版时间**:2015-06-23
基金项目:国家自然科学基金资助项目(61070234,61071167)
作者简介:赵礼峰(1959-),男,教授,硕士研究生导师,研究方向为图论及其在通信中的应用;梁 娟(1988-),女,硕士研究生,研究方向为图论及其在通信中的应用。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20150623.1051.044.html

有出弧的权值),再与 u_k 比较,较小的存入 u_k 中,从而简化了计算量。另一方面,改进算法通过引入路权数组有效地降低了算法的时间复杂度,同时借助前点标号数组增强了寻路直观性,实现了在更短时间内计算出指定节点到其余各节点最短路径的目的。

1　相关知识

定义 1　赋权图:给定有向图 $D=(V,A,W)$,如果 D 中任意一条边 (v_i,v_j) 上均有一个权 w_{ij},则称图 D 是一个赋权图[15]。

定义 2　回路:设 $D=(V,A,W)$ 是一个有向图,D 中含有一条起点和终点相同的路,则称为图 D 的一条回路。

定义 3　负回路:设 C 是 D 的一个回路,如果 $w(C)=\sum_{a\in A(C)} w(a)<0$,则称 C 为 D 的负回路。

定义 4　最短路:设 $D=(V,A,W)$ 是一个有向赋权网络,v_i 和 v_j 是 D 中的两个节点,D 的所有 (v_i,v_j) 路中权最小的路称为最短 (v_i,v_j) 路。

定理 1:设有向赋权网络 $D=(V,A,W)$,若 D 不含负回路,则 $\forall v_i,v_j\in V$,D 中最短有向 (v_i,v_j) 途径的权等于最短 (v_i,v_j) 路的权,并且最短 (v_i,v_j) 路的任何 (v_p,v_q) 节都是 D 中最短 (v_p,v_q) 路。反之,若 $\forall v_i,v_j\in V$,D 中最短有向 (v_i,v_j) 途径的权等于最短 (v_i,v_j) 路的权,则 D 不含负回路[13]。

定理 2:设有向赋权网络 $D=(V,A,W)$ [13],若 D 不含负回路,D 的自源节点 v_1 到节点 v_j 由不多于 k 条弧组成的路中权最小者记为 $P_j^{(k)}$,$1\leqslant j\leqslant n$,$1\leqslant k\leqslant n-1$,则 $u_j^{(k)}=w(P_j^{(k)})$ $(1\leqslant j\leqslant n,1\leqslant k\leqslant n-1)$,当且仅当 $u_j^{(k)}$ $(1\leqslant j\leqslant n,1\leqslant k\leqslant n-1)$ 满足下面的方程

$$\begin{cases} u_j^{(1)} = w_{1j}, 1\leqslant j\leqslant n \\ u_j^{(k)} = \min\limits_{1\leqslant i\leqslant n}\{u_i^{(k-1)}+w_{ij}\}, 1\leqslant j\leqslant n, 2\leqslant k\leqslant n-1 \end{cases}$$

2　改进算法

2.1　算法思想

改进算法的思想是逐次逼近,每次逼近都是求网络 D 中源节点 v_1 到其余各节点的带限制的最短路。先将自源节点 v_1 到节点 v_i 由一条弧组成的路的权值放在路权数组 U 中,再从当前路权数组 U 未圈起来的元素中找出最小的元素并将其圈起来,不妨记为 u_2,将 v_2 的所有出弧的权值分别加上刚圈起来的元素 u_2,并与 U 中对应的元素比较,较小的存入 U 中;再从当前数组 U 未圈起来的元素中找出最小的元素并将其圈起来,不妨记为 u_t,将 v_t 的所有出弧的权值分别加上刚圈起来的元素 u_t,并与 U 中对应的元素比较,较小的存入 U 中,直到 U 中的元素不再改变为止(此时 U 中的元素均已被圈起来)。路权数组 U 中被圈起来的元素为源节点到各节点的最短路长,同时根据前点标号数组反向追踪即可找到相应的最短路径。

2.2　算法步骤

设网络 D 的节点数是 n,弧数是 m。

Step1:把权矩阵 $(w_{ij})_{n\times n}$ 的第一行所有元素依次存放在数组 U 中,并将第一个元素 u_1 圈起来,将 $\boldsymbol{W}$ 第一列的所有元素都改为 $\times$,同时置前点标号数组 L(若 $u_i\neq\infty$,$l_i=1$,否则 $l_i=\infty$,$1\leqslant i\leqslant n$)。

Step2:从数组 U 未圈起来的元素中找出最小的元素 u_k,并把它圈起来,然后计算 u_k 加上 $\boldsymbol{W}$ 中第 k 行非零非 ∞的数 w_{ki},即 u_k+w_{ki}。

Step3:把 u_k+w_{ki} 与数组 U 中对应的元素(u_i 或 $\boxed{u_i}$)进行比较,较小的存放在数组 U 中(特别地,若 $u_k+w_{ki}<u_i$ 或 $u_k+w_{ki}<\boxed{u_i}$,则 $u_i\leftarrow u_k+w_{ki}$ 或 $\boxed{u_i}\leftarrow u_k+w_{ki}$,注意:若将要被替换掉的元素已被圈起来,则圈起来的标记也一同被替换掉。)。同时置 L 中 $l_i=k$,否则 l_i 不变。

Step4:如果数组 U 中的所有元素已被圈起来,则结束。数组 U 中圈起来的元素 u_i 表示最短 (v_1,v_i) 路的长,并可以根据前点标号数组 L 反向追踪找出最短 (v_1,v_i) 路,否则转 Step2。

2.3　算法的复杂度分析

改进算法的主要计算量是 Step2,计算当前数组 U 中最小的元素 u_k(即刚圈起来的元素)加上 w_{kj}(节点 v_k 的所有出弧的权值),当把所有节点的所有出弧都计算一遍时,计算量为 $O(m)$。但由于只计算权值变小的节点的所有出弧,所以每轮循环中(一轮循环,在算法步骤中体现为 $\boxed{u_i}\leftarrow u_k+w_{ki}$)很少出现每个节点权值都变小的情况,即计算量小于 $O(m)$,因为改进算法是逐次逼近算法,所以最多循环 $n-1$ 次。因此在最坏的情况下,算法的复杂度为 $O(mn)$。

2.4　算法的可行性分析

Ford 算法的可行性和正确性是由定理 2 保证的。文中的改进算法是在 Ford 算法的基础上,只计算权值变小的节点的所有出弧,对那些(与某个节点所有入弧相连的节点的权值均不变的)节点 v_i,计算后不会使该节点的权值变小,故可以不参与计算,从而减少了计算量。但与 Ford 算法本质没有太大区别。Step4 对算法作结束判断,若路权数组中的所有元素均已被圈起来,则算法结束。否则,继续迭代,由于 D 是无负回路网络,因此权值不可能永远变小,即 U 中的元素必

然会在有限步迭代后全被圈起来。所以,算法是可行的,也是正确的。

3 应用实例

在图 1 所示的带负权的有向图中,求从 v_1 到其余各节点的最短路。

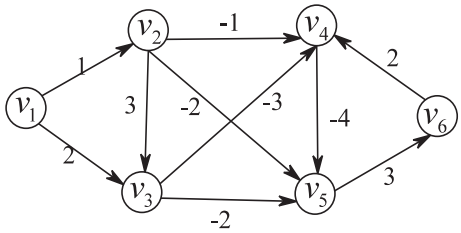

图 1 带负权的有向图

容易看出,该有向赋权图不含负回路,由图 1 得权矩阵如下:

$$\boldsymbol{W}=\begin{bmatrix}0&1&2&\infty&\infty&\infty\\ \infty&0&3&-1&-2&\infty\\ \infty&\infty&0&-3&-2&\infty\\ \infty&\infty&\infty&0&-4&\infty\\ \infty&\infty&\infty&\infty&0&3\\ \infty&\infty&\infty&2&\infty&0\end{bmatrix}$$

Step1: $U=[\boxed{0}\quad 1\quad 2\quad \infty\quad \infty\quad \infty]$

$$\boldsymbol{W}=\begin{bmatrix}\times&1&2&\infty&\infty&\infty\\ \times&0&3&-1&-2&\infty\\ \times&\infty&0&-3&-2&\infty\\ \times&\infty&\infty&0&-4&\infty\\ \times&\infty&\infty&\infty&0&3\\ \times&\infty&\infty&2&\infty&0\end{bmatrix}$$

$L=[1\quad 1\quad 1\quad \infty\quad \infty\quad \infty]$

Step2: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \infty\quad \infty\quad \infty]$,

$u_2+w_{23}=1+3=4$,

$u_2+w_{24}=1+(-1)=0$,

$u_2+w_{25}=1+(-2)=-1$

Step3: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad 0\quad -1\quad \infty]$,

$L=[1\quad 1\quad 1\quad 2\quad 2\quad \infty]$

Step4:因为数组 U 中有元素未被圈起来,所以转 Step2。

Step2: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad 0\quad \boxed{-1}\quad \infty]$,

$u_5+w_{56}=-1+3=2$

Step3: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad 0\quad \boxed{-1}\quad 2]$,

$L=[1\quad 1\quad 1\quad 2\quad 2\quad 5]$

Step4 因为数组 U 中有元素未被圈起来,所以转 Step2。

Step2: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \boxed{0}\quad \boxed{-1}\quad 2]$,

$u_4+w_{45}=0+(-4)=-4$

Step3: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \boxed{0}\quad -4\quad 2]$,

$L=[1\quad 1\quad 1\quad 2\quad 4\quad 5]$

Step4:因为数组 U 中有元素未被圈起来,所以转 Step2($\boxed{-1}\leftarrow u_4+w_{45}$,这是第一轮循环)。

Step2: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \boxed{0}\quad \boxed{-4}\quad 2]$,

$u_5+w_{56}=-4+3=-1$

Step3: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \boxed{0}\quad \boxed{-4}\quad -1]$,

$L=[1\quad 1\quad 1\quad 2\quad 4\quad 5]$

Step4:因为数组 U 中有元素未被圈起来,所以转 Step2。

Step2: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \boxed{0}\quad \boxed{-4}\quad \boxed{-1}]$,

$u_6+w_{64}=-1+2=1$

Step3: $U=[\boxed{0}\quad \boxed{1}\quad 2\quad \boxed{0}\quad \boxed{-4}\quad -1]$,

$L=[1\quad 1\quad 1\quad 2\quad 4\quad 5]$

Step4:因为数组 U 中有元素未被圈起来,所以转 Step2。

Step2: $U=[\boxed{0}\quad \boxed{1}\quad \boxed{2}\quad \boxed{0}\quad \boxed{-4}\quad \boxed{-1}]$,

$u_3+w_{34}=2+(-3)=-1$,

$u_3+w_{35}=2+(-2)=0$

Step3: $U=[\boxed{0}\quad \boxed{1}\quad \boxed{2}\quad -1\quad \boxed{-4}\quad \boxed{-1}]$,

$L=[1\quad 1\quad 1\quad 3\quad 4\quad 5]$

Step4:因为数组 U 中有元素未被圈起来,所以转 Step2($\boxed{0}\leftarrow u_3+w_{34}$,这是第二轮循环),在第五轮循环的 Step3 得到两个数组

$U=[\boxed{0}\quad \boxed{1}\quad \boxed{2}\quad \boxed{-1}\quad \boxed{-5}\quad \boxed{-2}]$

$L=[1\quad 1\quad 1\quad 3\quad 4\quad 5]$

Step4:因为数组 U 中所有的元素均已被圈起来,所以迭代结束。

数组 U 中圈起来的元素 u_i 表示最短 (v_1,v_i) 路的长,如: $u_5=-5$,表示最短 (v_1,v_5) 路的长是-5。

根据前点标号数组 L 还可以反向追踪找出最短 (v_1,v_i) 路,如: $l_5=4$ 表示节点 v_5 的前一个节点是 v_4,根据 $l_4=3$,表示节点 v_4 的前一个节点是 v_3,再根据 $l_3=1$,表示节点 v_3 的前一个节点是 v_1,这样,就可以找到最短 (v_1,v_5) 路径为 $v_1v_3v_4v_5$。

源节点 v_1 到其余各节点的最短路径和最短路长值见表 1。源节点 v_1 到其余各节点的最短路径见图 2。

4 仿真结果

为了验证改进算法的正确性和有效性,采用 Salama 提出的网络模型[16]进行实验。以下的仿真结

果均在 CPU 为 Inter Corei3 - 2330M,2.2 GHz 内存 2 GB,MATLAB7.9 环境下完成。比较用改进算法和 Ford 算法计算规模分别为 100 阶,200 阶,直至 2 000 阶网络中从源节点到其余各节点的最短路的运行时间,数据结果如表 2 所示。

表 1　源节点 v_1 到其余各节点最短路径与最短路长值

节点	最短路径	路长
1,2	①→②	1
1,3	①→③	2
1,4	①→③→④	-1
1,5	①→③→④→⑤	-5
1,6	①→③→④→⑤→⑥	-2

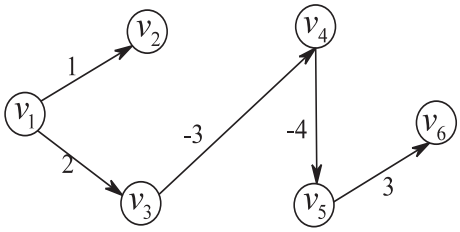

图 2　源节点 v_1 到其余各节点的最短路径

表 2　对不同规模网络改进算法与原算法的运行时间　　s

网络规模	改进算法	Ford 算法
100	0.010 9	0.056 2
200	0.030 9	0.283 2
300	0.050 2	0.899 3
400	0.073 3	2.161 6
500	0.101 3	4.348 5
600	0.138 1	7.937 1
700	0.181 3	13.433 5
800	0.232 6	19.323 3
900	0.291 1	28.839 9
1 000	0.350 6	37.792 9
1 100	0.427 5	52.230 9
1 200	0.504 9	65.005 1
1 300	0.600 1	85.892 1
1 400	0.692 9	103.542 5
1 500	0.794 1	148.927 3
1 600	0.894 6	174.977 5
1 700	1.007 4	207.843 3
1 800	1.136 7	237.150 6
1 900	1.289 7	304.282 2
2 000	1.423 2	356.024 2

图 3 给出了两种算法下计算源节点相同的同一网络最短路的运行时间。

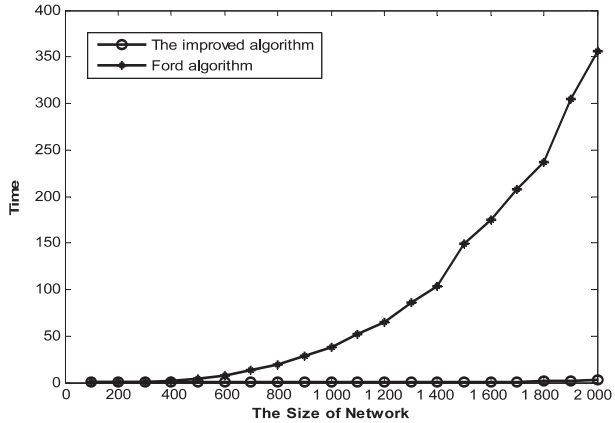

图 3　不同规模网络改进算法与原算法的运行时间比较图

通过仿真比较,用改进算法和原算法这两种算法分别求解最短路径的路长值与最短路径,所求到的结果都相同,但改进算法所用的计算时间明显比 Ford 算法少。并且从图像上可以看出:随着网络规模的增大,改进算法比 Ford 算法能更快地找出网络中从源节点到其余各节点的最短路径,是一种更为有效的算法。

5　结束语

无负回路网络的最短路问题已被广泛应用于生产生活的诸多领域,Ford 算法是求解从源节点到其余各节点最短路问题的经典算法。文中的改进算法是在 Ford 算法的基础上,只计算权值变小的节点的所有出弧,对于 Ford 算法中与某个节点所有入弧相连节点的权值均不变的节点,都不参与计算,从而简化了计算量。另外,改进算法引入路权数组和前点标号数组,不仅有效地降低了算法的时间复杂度,而且增强了寻路直观性,实现了在更短时间内计算出从源节点到其余各节点最短路的目的。

参考文献:

[1] Festa P, Guerriero F, Laganà D, et al. Solving the shortest path tour problem[J]. European Journal of Operational Research, 2013,230(3):464-474.

[2] Loachim I. A dual programming algorithm for the shortest path problem[J]. Networks,1998,31(2):193-204.

[3] 孙小军,刘三阳,焦建民. 基于最小树权矩阵法的改进算法[J]. 计算机工程与设计,2005,26(12):3274-3275.

[4] Kuipers F, Mieghem P V, Korkmaz T, et al. An overview of constraint- based path selection algorithms for QOS routing [J]. IEEE Communication Magazine,2002,40(12):50-55.

[5] Cormen T H, Leiserson C E, Rivest R L. Introduction to algorithms[M]. 2nd ed. Cambrige, USA: MIT Press,2001.

[6] Xin Lu, Camitz M. Finding the shortest paths by node combination[J]. Applied Mathematics and Computation,2011,217

(下转第 81 页)

CCA 可以很好挖掘不同模态之间的相关性,能够达到更好的检索效果。

表 1　CCA 和 SVD 两种方法所得的总复相关性比较

前 8 个组合变量	CCA 方法		SVD 方法	
	R	R'	R	R'
1	0.981 3	0.981 3	0.981 2	0.981 2
2	0.975 6	0.975 6	0.973 4	0.977 4
3	0.971 5	0.971 5	0.951 6	0.956 2
4	0.962 9	0.962 9	0.933 4	0.946 9
5	0.951 2	0.951 2	0.884 7	0.901 4
6	0.948 1	0.948 1	0.853 2	0.898 1
7	0.934 1	0.934 1	0.738 2	0.806 6
8	0.917 3	0.917 3	0.533 3	0.633 1

6　结束语

目前大部分的音乐搜索系统仍然沿用传统的基于关键字匹配的检索模式,这种检索机制仅仅是基于文本进行查询匹配,忽视了音乐本身的底层特征,导致查准率差强人意,越来越难以满足人们日益个性化和智能化的需求。文中提出并实现了一种跨模态检索方式,考虑多个模态的特性,让各模态之间互相补充信息,来提高检索准确率。当然该方式也有其不足的地方,比如当用户提交的关键词不在数据库中时就会需要对用户输入的自然语言进行分词分析和词性标注,并进行语义分析,选取相似度最高的数据库中的语义作为检索扩展词进行检索[15]。下一步工作的重点就是如何能够直接将用户提交的检索词映射到 CCA 子空间。

参考文献:

[1] 梅　放,林鸿飞. 基于社会化标签的移动音乐检索[C]//第五届全国信息检索学术会议论文集. 出版地不详:出版者不详,2009.

[2] 王发友. 典型相关分析的基本思想和方法步骤[J]. 科技信息:学术研究,2007(36):472-472.

[3] Rabiner L R, Juang B H. Fundamentals of speech recognition [M]. Englewood Cliffs: PTR Prentice Hall, 1993.

[4] Kim H G, Sikora T. Audio spectrum projection based on several basis decomposition algorithms applied to general sound recognition and audio segmentation[C]//Proc of EUSIPCO. [s. l.]:[s. n.], 2004:1047-1050.

[5] Herrera P, Yeterian A, Gouyon F. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques [M]//Music and artificial intelligence. Berlin: Springer, 2002:69-80.

[6] Mandel M I, Ellis D P W. Song-level features and support vector machines for music classification[C]//Proc of 6th international conference on music information retrieval. London: [s. n.], 2005:594-599.

[7] 汪　海. 一种基于 RASTA-PLP 分析的话者识别技术[J]. 电声技术,2002(5):8-9.

[8] Kingsbury B E D, Morgan N. Recognizing reverberant speech with RASTA-PLP[C]//Proc of 1997 IEEE international conference on acoustics, speech, and signal processing. [s. l.]: IEEE, 1997:1259-1262.

[9] Zhen Bin, Wu Xihong, Liu Zhimin, et al. An enhanced relative spectral processing of speech[J]. Chinese Journal of Acoustics, 2002, 21(1):86-96.

[10] 梁胜杰,张志华,崔立林. 主成分分析法与核主成分分析法在机械噪声数据降维中的应用比较[J]. 中国机械工程,2011,22(1):80-83.

[11] 范丽伟,唐焕文,唐一源. 独立成分分析应用于 fMRI 数据研究[J]. 大连理工大学学报,2003,43(4):399-402.

[12] 钱国华,荀鹏程,陈　峰,等. 偏最小二乘法降维在微阵列数据判别分析中的应用[J]. 中国卫生统计,2007,24(2):120-123.

[13] 严华生,王学仁. 多因变量及要素场统计预报[M]. 北京:气象出版社,1991.

[14] 严华生,尤卫红. 多因变量矩阵回归预报方法[J]. 大气科学,1997,21(4):493-498.

[15] 孙　亮,任小康. 基于本体的图像语义检索模型[J]. 重庆工学院学报:自然科学版,2009,23(1):127-131.

(上接第 75 页)

(13):6401-6408.

[7] Kamiński M, Medvedev P, Milanič M. Shortest paths between shortest paths[J]. Theoretical Computer Science, 2010, 412(39):5205-5210.

[8] 龚　劬. 图论与网络最优化算法[M]. 重庆:重庆大学出版社,2009:48-54.

[9] 王　磊. 基于 Dijkstra 算法的多目标城市公交最优化查询的快速算法[J]. 信息通信,2012(6):65-67.

[10] 孙小军. 最短路问题的改进算法[J]. 计算机工程与设计,2009,30(16):3762-3764.

[11] 霍丽娜,刘三阳,刘　磊. 一种改进的 Ford 算法[J]. 现代电子技术,2007,30(20):111-113.

[12] 韩伟一. 经典 Bellman-Ford 算法的改进及其实验评估[J]. 哈尔滨工业大学学报,2012,44(7):74-77.

[13] 谢　政. 网络算法与复杂性理论[M]. 长沙:国防科技大学出版社,2003:79-93.

[14] 杨　雷. 单源最短路问题高效算法探究[J]. 电脑知识与技术,2009,5(13):3439-3442.

[15] 赵礼峰,蒋腾飞. 无回路网络最短路径的一种新算法[J]. 计算机技术与发展,2013,23(2):105-107.

[16] 周　灵. Waxman-Salama 模型网络拓扑生成算法设计与实现[J]. 湖南理工学院学报,2008,21(2):40-42.

Ford算法的改进算法

作者： 赵礼峰， 梁娟， ZHAO Li-feng， LIANG Juan
作者单位： 南京邮电大学 理学院,江苏 南京,210023
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2015(7)