第41卷 第6A期 计 算 机 科 学 Vol. 41 No. 6A
2014年6月 Computer Science June 2014

两区域交叉网络图的 Dijkstra 改进算法

阳西述[1] 刘怀玉[2] 胡亚辉[3]
(湖南第一师范学院信息科学与工程系 长沙 410205)[1] (湖南第一师范学院教育科学系 长沙 410205)[2]
(湖南第一师范学院数学系 长沙 410205)[3]

摘 要 传统 Dijkstra 算法是计算网络图单源最短路径的经典算法,但不适应于现实中存在的两区域交叉网络图。提出了新的区域特征码概念,设计了两区域交叉网络图的区域特征码和访问控制逻辑,并以此为基础改进了 Dijkstra 算法。实验证明,改进以后的 Dijkstra 算法能正确地计算两区域交叉网络图的单源最短路径,其时、空复杂度与原算法相同。通过这种改进,扩展了 Dijkstra 算法的适应范围。

关键词 Dijkstra 算法,两区域交叉网络图,区域特征码,访问控制逻辑
中图法分类号 TP391 **文献标识码** A

Improved Dijkstra Algorithm for Two Regional-cross Network Diagram

YANG Xi-shu[1] LIU Huai-yu[2] HU Ya-hui[3]
(Department of Information Science and Engineering, Hunan First Normal University, Changsha 410205, China)[1]
(Department of Education Science, Hunan First Normal University, Changsha 410205, China)[2]
(Department of Maths, Hunan First Normal University, Changsha 410205, China)[3]

Abstract The traditional Dijkstra algorithm is the classical algorithm to calculate the single-source shortest path for a network diagram, but not suited for two regional-cross network diagrams in reality. In this paper, the new concept of regional signature was put forward, and the regional signature and access-control logic for two regional-cross network diagram were designed too. Based on these, the Dijkstra algorithm was improved. The experiment proved that the improved Dijkstra algorithm can correctly calculate the single-source shortest path for two region-cross network diagrams, its place complexity & time complexity are the same as the traditional algorithm. Through this improvement, the adaptation range of the Dijkstra algorithm was expanded.

Keywords Dijkstra algorithm, Two regional-cross network diagram, Regional signature, Access-control logic

1 引言

Dijkstra 算法是关于有向(或无向)网络图单源最短路径的经典算法,该算法已广泛应用于计算机网络、地理信息系统、城市交通管理、社会复杂网络等诸多领域[1-4]。专业人员为了提高 Dijkstra 算法效率,做了许多研究,其中一类使用图的邻接表(或十字交叉链表)取代图的邻接矩阵来存储图的结点信息[5],另一类使用排序优先队列或排序堆来代替传统算法中的无序表[6,7]。前者通过降低图的存储空间来提高 Dijkstra 算法的效率,后者通过结点排序来减少查询时间的复杂度,从而提高算法的效率。但是传统 Dijkstra 算法以及这些改进措施,都是针对网络图内各个结点之间相互访问没有限制(单一区域网络图)的情况。设一个网络图有两个区域,两区域又相互交叉(如图 1 所示),规定同一区域内结点可相互访问,交叉区域的结点可被两个区域访问,而两区域非交叉部分的结点不能互相访问。对于这样的网络图,要计算从某一结点到达其它各结点的单源最短路径,Dijkstra 算法要作怎样的改进?目前国内外尚未见相关研究成果,本文将研究这个问题。

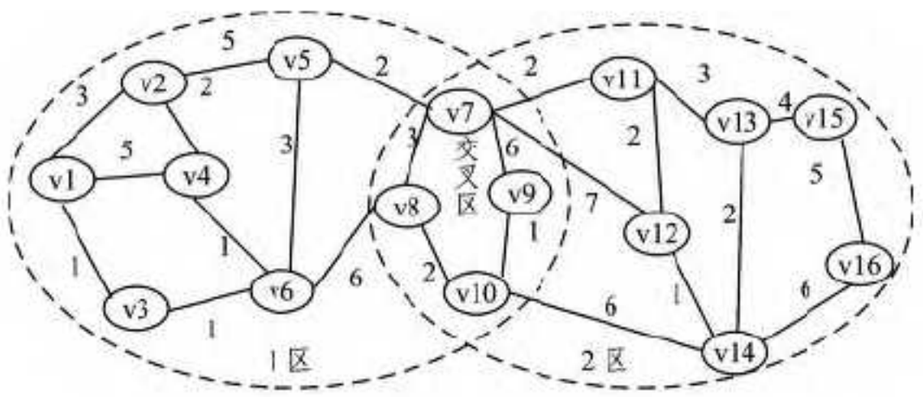


图 1 两区域交叉网络图

2 现实中的两区域交叉网络图

两区域交叉网络图广泛存在于技术和社会领域。在计算机网络领域,例如一个学校拥有校园网络,校园网络内一些结点(如教育视频会议的结点)同时又属于另一个网络——城域教育视频会议网络。校园网络和城域教育视频会议网络可看成一个网络图中的两个区域,而校内视频会议有关结点是两

本文受湖南省教育科学重大项目(XJK011DDUT003),湖南省科技计划项目(2012TZ2018,2013SK3137),湖南第一师范学院项目(XYS10Z07,XYS11Z06),计算机网络精品课程项目资助。
阳西述(1966—),男,硕士,教授,主要研究方向为网络工程、并行计算,E-mail:hncsyxs@163.com;刘怀玉(1966—),女,高级讲师,主要研究方向为数学教育;胡亚辉(1965—),男,博士,教授,主要研究方向为组合数学。

区域的交叉区。按照信息安全有关规定，校园网里的结点可相互访问，城域教育视频会议网络内结点也能相互访问，校内有关视频会议的结点（交叉区结点）则能同时被两个区域访问。但是，校园网络的其它结点却不允许访问城域教育视频会议网络的结点，城域教育视频会议网络的结点也不允许访问校园网的其它结点。这是计算机网络领域存在的两区域交叉网络图。

其它领域也存在两区域交叉网络图。例如地理信息系统中有军用信息系统、民用信息系统，还有军民两用信息。在道路交通与航空领域，各国有各自的国内交通路网与航线网，邻国之间可能有双方都能通行的公共交通路区域或航空区域（缓冲区）。在社会复杂网络等中，也有类似的情况。

定义 1（两区域交叉网络图） 如果一个网络图由相互交叉的两个区域组成，单个区域内结点能相互访问，两区域之间非交叉部分的结点不能相互访问，交叉部分的结点能被两个区域访问，这样的网络图就叫做两区域交叉网络图。

传统 Dijkstra 算法[8] 以及现有的各种改进方法[4-7] 都没有考虑网络图分为两个或多个区域、区域相互交叉的情况，必须修改 Dijkstra 算法才能使其应用于两区域交叉网络图。

3 传统 Dijkstra 算法

传统 Dijkstra 算法计算网络图单源最短路径的步骤[8] 如下：

(1) 用邻接矩阵 $G[N][N]$ 表示网络图 $G(V,E)$，$G[i][j]$ 表示边 (v_i,v_j) 上的权值，若不存在边 (v_i,v_j)，则置 $G[i][j]$ 为 ∞（计算机上用一个最大值代替）。S 为已找到的从初始结点 x 出发的最短路径的终点集合，它的初始状态为空集。一维数组 $Dist[N]$ 保存从 x 到达其余各点的最短路径长度，初始时

$$Dist[i]=G[x][i]$$

(2) 选择 v_j，使得

$$Dist[j]=\min\{Dist[i]\mid v_i\in V-S\}$$

v_j 就是当前求得的一条从 x 出发的最短路径的终点。将 v_j 加入已标记顶点集中：

$$S=S\cup\{v_j\}$$

(3) 修改从 x 出发到集合 $V-S$ 上任一顶点 v_k 可达的最短路径长度。如果

$$Dist[j]+G[j][k]<Dist[k]$$

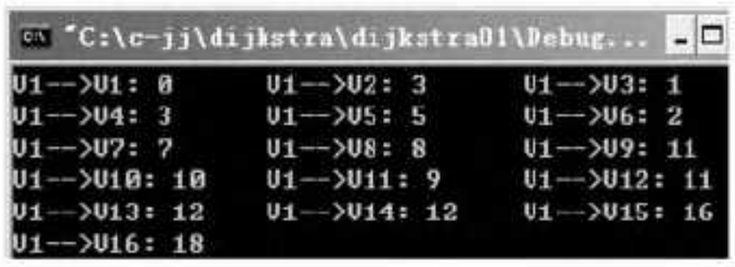
则修改 $Dist[k]$ 为

$$Dist[k]=Dist[j]+G[j][k]$$

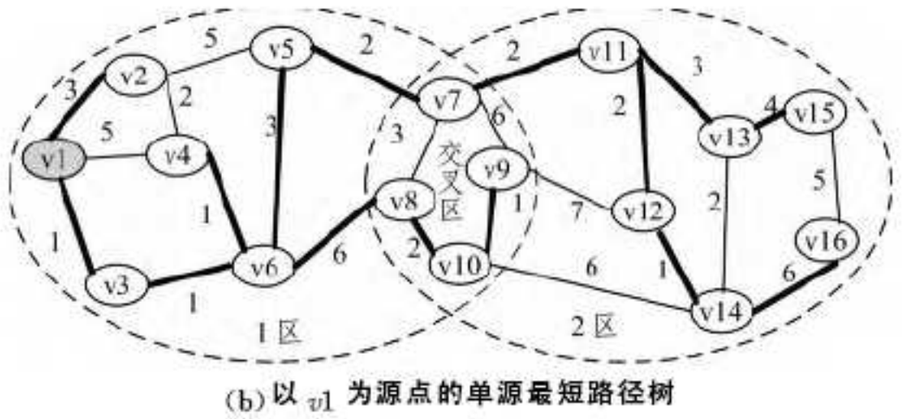
(4) 重复操作(2)、(3)共 $N-1$ 次。由此求得从顶点 x 出发到达图上其余各顶点的最短路径。

应用传统 Dijkstra 算法来计算图 1 所示的两区域交叉网络图，例如计算从源点 $v1$ 到达其余各顶点的单源最短路径值（用 C 语言编程实现），如图 2(a) 所示，相应的最短路径树如图 2(b) 所示。

从图 2 所示实验结果可看出，用传统 Dijkstra 算法来计算两区域交叉网络图的单源最短路径时，从源点 $v1$ 出发，能够访问到图中所有的结点，这与两区域交叉网络图的定义相矛盾，按照定义 1 的规定，1 区的 $v1$ 结点不能访问 2 区的 $v11-v16$ 结点（即从 $v1$ 到 $v11-v16$ 的路径值应为 ∞）。所以计算结果有错误。必须改进 Dijkstra 算法，才能使其适用于两区域交叉网络图。



(a) 从 $v1$ 点到达其余各顶点的单源最短路径值



(b) 以 $v1$ 为源点的单源最短路径树

图 2 传统 Dijkstra 算法计算的单源最短路径

4 Dijkstra 改进算法

4.1 区域特征码与访问控制逻辑

由于两区域交叉网络图不同区域之间的结点访问受到限制，为达到通过简易数值计算就能判定结点之间能否相互访问的目标，我们先给两区域交叉网络图的区域进行特征编码。

定义 2（区域特征码） 给两区域交叉网络图的每个区域编一个不同的二进制特征数，让每一对结点所在区域的二进制特征数作某种逻辑运算，根据运算结果就能判断结点之间能否相互访问，这样的二进制特征数就叫区域特征码。

定义 3（访问控制逻辑） 将网络图任意两个结点所在区域的区域特征码作按位与（&）的逻辑运算，若结果为 0 则表示两结点不能相互访问，若结果非 0 则表示两结点可以相互访问，这样的逻辑就叫访问控制逻辑。

按照定义 2 和定义 3 的规定，我们设计如图 1 所示的两区域交叉网络图的区域特征码，网络图共有 3 个区域（1 区、2 区和交叉区），区域特征码可以设计成 2 位二进制数，如表 1 所列。将区域特征码两两相互作按位与（&）的逻辑运算得到的访问控制逻辑结果如表 2 所列。

表 1 两区域交叉网络图的区域特征码

区域	1 区	2 区	交叉区
区域特征码	01	10	11

表 2 区域特征码作按位与（&）逻辑运算的访问控制逻辑

& 运算	01(1 区)	10(2 区)	11(交叉区)
01(1 区)	1	0	1
10(2 区)	0	10	10
11(交叉区)	1	10	11

表 2 中只有 1 区和 2 区的区域特征码相互按位与逻辑运算的结果为 0（即：01 & 10=0、10 & 01=0），其它情况都是非 0。根据访问控制逻辑的定义（定义 3），1 区和 2 区的区域特征码按位与逻辑运算结果为 0，表示此两区域的结点不能互相访问；其它情况（1 区和 1 区、1 区和交叉区、2 区和 2 区、2 区域和交叉区）区域特征码按位与逻辑运算的结果非 0，表示对应两区域的结点可以相互访问。这个结论完全符合两区域交叉网络图的定义（定义 1），说明表 1 所设计的两区域交叉网络图的区域特征码是正确的。

4.2 Dijkstra 改进算法

以区域特征码和访问控制逻辑为基础，改进了 Dijkstra

算法。适应于两区域交叉网络图的 Dijkstra 改进算法如下。

(1)用邻接矩阵 $G[N][N]$ 存储网络图 $G(V,E)$，$G[i][j]$ 表示边 (v_i,v_j) 上的权值，若不存在边 (v_i,v_j)，则置 $G[i][j]$ 为 ∞。S 为已找到的从 x 出发的最短路径的终点集合，它的初始状态为空集。一维数组 $Area[N]$ 用于保存结点 v_i 所在区域的特征码。一维数组 $Dist[N]$ 保存从 x 到达其余各点的最短路径长度，初始时根据两结点所在区域的访问控制逻辑值 $(Area[x]\&Area[i])$ 来判断 x 能否访问 v_i，若能，则 $Dist[i]$ 初值为 $G[x][i]$，否则为 ∞。即

$$Dist[i]=Area[x]\&Area[i]\ ?\ G[x][i]:\infty$$

(2)选择 v_j，使得

$$Dist[j]=\min\{Dist[i]\mid v_i\in V-S\}$$

v_j 即 $V-S$ 集合里从 x 出发的一条最短路径的终点。将 v_j 加入已标记顶点集里：

$$S=S\cup\{v_j\}$$

(3)修改从 x 出发到集合 $V-S$ 里可访问顶点 v_k（$Area[x]\&Area[k]$ 是否非 0）的最短路径长度。如果

$$Area[x]\ \&\ Area[k]\ !=0\ \wedge\ Dist[j]+G[j][k]<Dist[k]$$

则 $Dist[k]$ 修改为

$$Dist[k]=Dist[j]+G[j][k]$$

(4)重复操作(2)、(3)共 $N-1$ 次。由此求得从 x 出发到图上其余各可达顶点的最短路径。

Dijkstra 改进算法 C 语言源码如下：

```
void Dijkstra(int G[][],int area[],int dist[],int x)
  { int i,j,k,v,min,s[N];
  for (i=0;i < N;i++) // 初始化
  {s[i]=False;//初始时每个 vi 都未加入 S
  dist[i]=area[x] & area[i] ?G[x][i]:MAX;
  /*初始化 dist[i]，若 x 能访问 i，则 dist[i]等于 G[x][i]，否则为∞
     */
}
  for (i=0;i < N;i++)
    //计算单源最短路径
      { min=MAX;//min 初始值为∞
    for (v=0;v < N;v++)
      // 在 V-S 中寻找距离 x 最近的顶点
        { if ( !s[v] && dist[v] < min)
          { min=dist[v];
            j=v;
          }
      }
    s[j]=True;// 将结点 j 加入 S
    for (k=0;k < N;k++)
      // 更新 V—S 中可访问顶点的 dist[]值
    if ( !s[k] && (area[x]&area[k]) && (dist[j]+G[j][k]<dist
[k]))
          dist[k]=dist[j]+G[j][k];
  }
}
  //有关网络图 G[][]、区域特征码 area[]数组的初始化，以及结点和最短路径值 Dist[]的显示等(略)
```

4.3 Dijkstra 改进算法实验

将每个区域的特征码(见表 1)标注到两区域交叉网络图上，如图 3 所示。

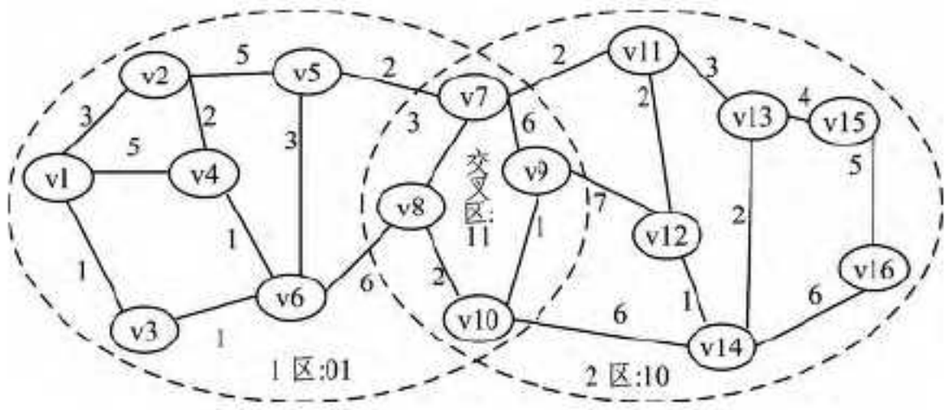
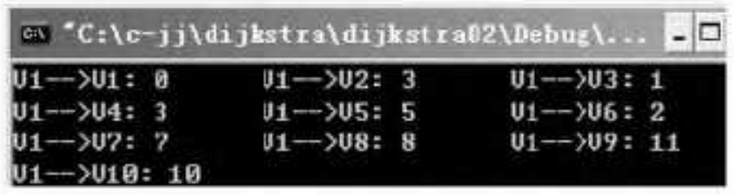
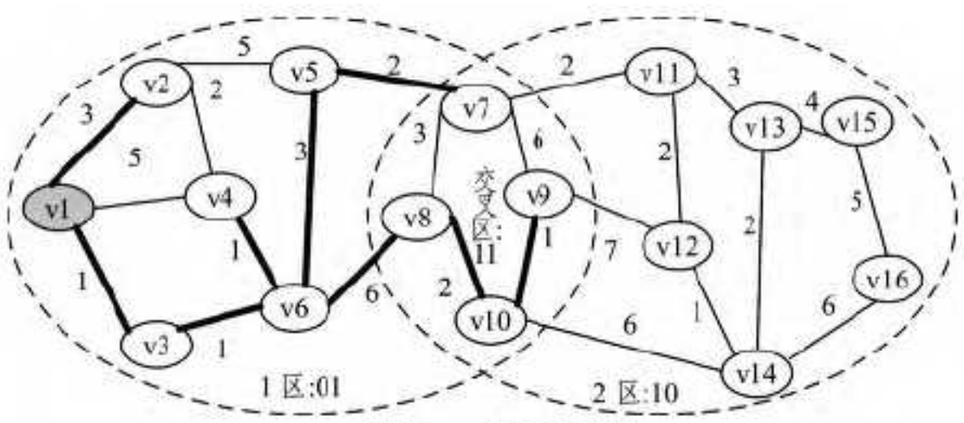


图 3 标注了特征码的两区域交叉网络图

对图 3 运行 Dijkstra 改进算法 C 语言程序(C 语言程序见附件)。以 1 区域的 $v1$ 为源点，计算出来的单源最短路径值如图 4(a)所示，路径不可达(值为 ∞)的未列出，相应的单源最短路径树如图 4(b)所示。



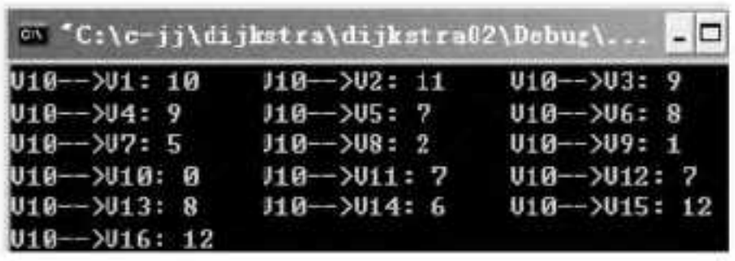
(a)单源($v1$)最短路径值



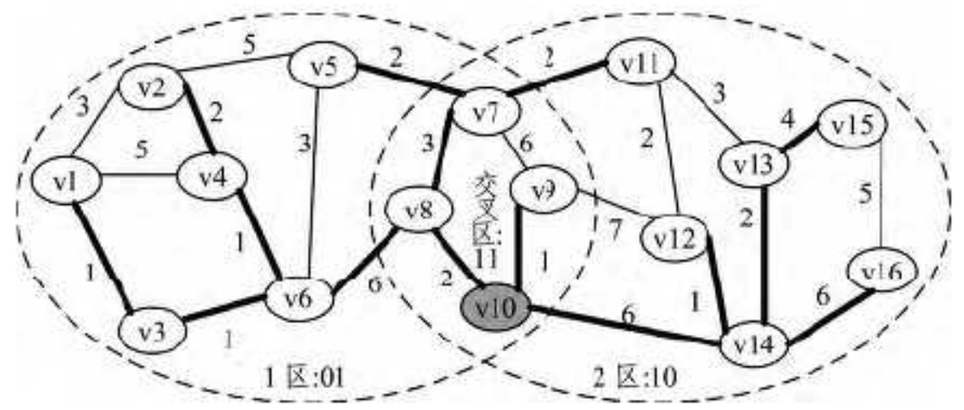
(b)单源($v1$)最短路径树

图 4 Dijkstra 改进算法计算的以 $v1$ 为源点的单源最短路径

若以交叉区域的 $v10$ 点为源点，则计算出来的单源最短路径值(路径不可达的未列出)及相应的单源最短路径树如图 5(a)、(b)所示。



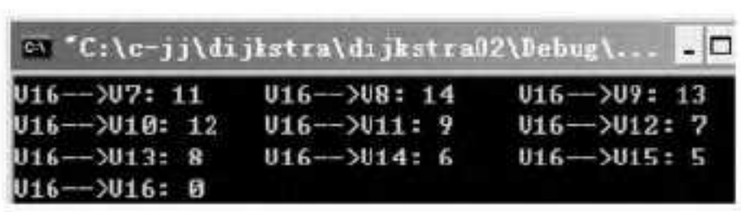
(a) 单源($v10$)最短路径值



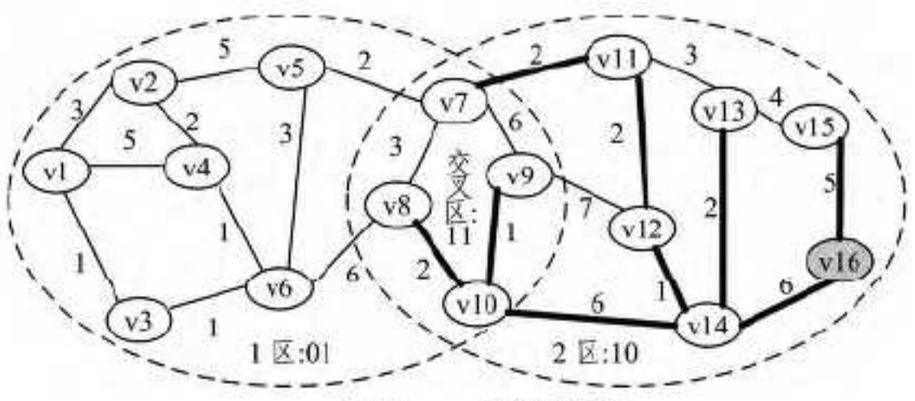
(b) 单源($v10$)最短路径树

图 5 Dijkstra 改进算法计算的以 $v10$ 为源点的单源最短路径

若以 2 区域的 $v16$ 点为源点，则计算出来的单源最短路径值(路径不可达的未列出)及相应的单源最短路径树如图 6(a)、(b)所示。



(a) 单源($v16$)最短路径值



(b)单源($v16$)最短路径树

图6 Dijkstra改进算法计算的以$v16$为源点的单源最短路径

从以上实验结果可知,1区域的结点只能访问1区域和交叉区域的结点,不能访问2区域的结点(见图4);交叉区域的结点,能访问交叉区域、1区域和2区域的所有结点(见图5);2区域的结点只能访问2区域和交叉区域的结点,不能访问1区域的结点(见图6)。这完全符合两区域交叉网络图结点之间的访问限制规则(见定义1)。因此,无论源点在两区域交叉网络图的哪一个区域,采用本文所述Dijkstra改进算法都能正确地计算出它到达其余各点的单源最短路径。

5 分析比较

从正确性、空间复杂度和时间复杂度3方面来比较传统Dijkstra算法与本文改进Dijkstra算法应用于两区域交叉网络图的异同。

首先分析算法的正确性。使用传统Dijkstra算法计算两区域交叉网络的单源最短路径时,由于算法不能识别结点所在区域和访问控制逻辑的限制,从本文第3节的实验计算结果可知,用传统Dijkstra算法计算出来的两区域交叉网络图的单源最短路径是错误的(图2中从结点$v1$到$v11-v16$的最短路径结果错误)。文献[17]中的改进Dijkstra算法由于同样的原因,也不能正确计算两区域交叉网络图的单源最短路径。而本文提出的Dijkstra改进算法是基于区域特征码和访问控制逻辑所构建的,能够识别结点所在区域和访问控制逻辑。从第4.3节的实验结果(图4—图6)可知,本文提出的Dijkstra改进算法能够正确地计算出两区域交叉网络图从任一结点出发的单源最短路径。

其次,分析空间复杂度。传统Dijkstra算法用一个二维数组$G[N][N]$来存储网络图,其空间复杂度是$O(N^2)$[9]。本文所述Dijkstra改进算法,在存储空间方面增加了一个一维数组$area[N]$,用来保存各结点所在区域的特征码,其空间复杂度是$O(N^2+N)\approx O(N^2)$。因而本文Dijkstra改进算法与传统Dijkstra算法的空间复杂度具有相同数量级,都是$O(N^2)$。

再分析时间复杂度。本文设计的Dijkstra改进算法与传统Dijkstra算法的运行时间差别主要有两方面的因素:一是改进算法的第(1)步在初始化$dist[i]$前,增加了一个判断条件——x能否访问v_i(即$area[x]\&area[i]$是否非0);第(3)步在更新$V-S$中结点的$dist[k]$值之前,也增加了这样的判断条件。增加了判断条件,会稍微增加运算的时间。二是改进算法中这两处判断条件增加以后,使得第(3)步中$V-S$集合里有些结点的$dist[]$值不需要更新,这就节省了一些运算时间,具体节省多少时间与网络图的结点数以及网络图交叉部分结点所占比例有关。这两个因素分别增加和减少了改进算法的运行时间。综合这两个方面,改进后的算法时间复杂度与原算法相差不大。我们在一台Intel3.3G*2CPU和4GB内存的PC机上,构造了90000个结点的两区域交叉网络图,通过运行Dijkstra改进算法C语言程序,计算从一个结点到其它所有结点的单源最短路径,平均耗时约3.44ms;对同一个图运行传统Dijkstra算法C语言程序(尽管传统算法运算结果不正确,这里只需要用它的运算时间作比较),平均耗时约3.35ms。当改变图的结构或者结点数目以后,运算时间差别有所不相同,但两个算法用时相差不大。从算法结构上分析,传统Dijkstra算法计算一个网络图的单源最短路径时,使用二重$0\sim N-1$循环,其时间复杂度是$O(N^2)$[9];本文提出的Dijkstra改进算法也是使用二重$0\sim N-1$循环,执行步骤与循环层次与传统Dijkstra算法相同,其时间复杂度也是$O(N^2)$。

文献[6,7]提出的排序堆或堆配对Dijkstra算法,计算单一区域网络图的单源最短路径的时间复杂度是$E+N\log(N)$。若用本文提出的区域特征码和访问控制逻辑去改造排序堆或堆配对Dijkstra算法,用于计算两区域交叉网络图的单源最短路径时,其时间复杂性也将是$E+N\log(N)$。

结束语 两区域交叉网络图广泛存在于现实中,传统Dijkstra算法不能用于计算两区域交叉网络图。本文独创性地提出了区域特征码的概念和访问控制逻辑,并以此为基础改造了Dijkstra算法,从而解决了传统Dijkstra算法不能正确计算两区域交叉网络图的单源最短路径问题。改进后的Dijkstra新算法可以正确地计算出两区域交叉网络图中任一源点出发到达其它各点的单源最短路径。改进算法的空间复杂度、时间复杂度与传统算法相当。通过本文的研究,扩大了Dijkstra算法的应用范围,为实际应用Dijkstra改进算法来计算两区域或多区域交叉网络图的单源最短路径奠定了算法基础。

参考文献

[1] 谢希仁.计算机网络(第5版)[M].北京:电子工业出版社,2011(9):152-156

[2] Xie De-xiang,Zhu Hai-bo,Yan Lin,et al. An improved Dijkstra algorithm in GIS application[C]//CDC'2010. 2010:167-169

[3] 王峰,游志胜,曼丽春,等.Dijkstra及基于Dijkstra的前N条最短路径算法在智能交通系统中的应用[J].计算机应用研究,2006(9):203-206

[4] Cuan Ying,Chen Xiao-ni. Application of Improved Dijkstra Algorithm in Selection of Gas Source Node in Gas Network[C]// 2012 International Conference on Industrial Control and Electronics Engineering. 2012,410:1558-1560

[5] Zhan F B. Three Fastest Shortest Path Algorithms on Real Road Networks[J]. Journal of Geographic Information and Decision Analysis,1997,1(1):69-82

[6] 李元臣,刘维群.基于Dijkstra算法的网络最短路径分析[J].微计算机信息,2004(3):295-298

[7] 张林广,方金广,申排伟.基于配对堆的Dijkstra算法[J].中国图形图象学报,2007(5):922-926

[8] 严蔚敏,吴伟民.数据结构(C语言版)[M].北京:清华大学出版社,2002(9):187-190

[9] Thomas H,Cormen C E,Leiserson R L. Rivest Clifford Stein, Introduction to Algorithms(Second Edition)[M]. MIT Press, 2001,China Machine Press,2006:366-369