

# 基于邻接表的图生成算法探讨

董加强

(西昌学院,四川 西昌 615013)

**【摘要】**本文详细介绍了几类重要的图及其生成算法,包括随机图、K-邻接图、欧几米德邻接图。对于每一类图都用C语言描述其产生算法,并对算法的性能进行了简单的分析,同时对每一个算法用一组实际数据进行了演示,得到了具体的图。

**【关键词】**邻接表;图;算法

**【中图分类号】**TP391.41 **【文献标识码】**A **【文章编号】**1673-1891(2009)02-0043-03

图的生成算法是图论研究的一个重要方面,为了对这些算法进行试验、测试和研究,就必须根据实际需要生成各种不同类型的图。针对这种需要,本文对常用的几类图及其生成算法进行了探讨,并给出具体的图。

## 1 图的邻接表存储结构

用 $G(V,E)$ 来表示一个具有 $V$ 个顶点和 $E$ 条边的图,用 $0 \sim V-1$ 对顶点进行编号, $v-w$ 表示顶点 $v$ 到顶点 $w$ 的边。图的存储结构有邻接矩阵和邻接表<sup>[1]</sup>。本文探讨的图生成算法均采用如图1所示的邻接表作为其存储方式。

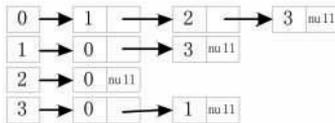


图1 邻接表

图的邻接表实现需要使用一个链表数组,一个链表对应一个顶点,边 $v-w$ 由链表 $v$ 中 $w$ 对应的结点以及链表 $w$ 中 $v$ 对应的结点表达。

```

typedef struct node *link;
typedef struct{int v; int w;} Edge;
struct node{int v; link next;};
struct Graph{int V; int E; link *adj;};
Edge EDGE(int v, int w)
{ Edge e;
e.v = v; e.w = w; return e;
}
link NEW(int v, link next)
{ link x = (link)malloc(sizeof(*x));
x->v = v; x->next = next; return x;
}
Graph GRAPHinit(int V)
{ int v; Graph G;
G.V = V; G.E = 0;
G.adj = (link *)malloc(V*sizeof(link));

```

```

for (v = 0; v < V; v++) G.adj[v] = NULL;
return G;
}
void GRAPHinsertE(Graph G, Edge e)
{ int v = e.v, w = e.w;
G.adj[v] = NEW(w, G.adj[v]); G.adj[w] =
NEW(v, G.adj[w]);
G.E++;
}

```

该算法生成的是多图,它没有删除平行边,而且还存在自循环。检查邻接表中的重复边必须搜索链表,所花时间与 $V$ 成正比。同时删除边时必须搜索两个链表,找到要删除的结点,时间与 $V$ 成正比。所以对于稠密图,在时间上的开销较大,不适合大量删除边的情况。

## 2 几类图及其生成算法探讨

### 2.1 随机图

这种方法通过生成 $E$ 个随机整数对 $v-w$ ( $v, w=0, 1, 2, \dots, V-1$ )来生成 $E$ 条随机边。

```

int RANDv(Graph G)
{ return (rand() % G.V); }
Graph GRAPHrandE(int V, int E)
{ Graph G = GRAPHinit(V);
while(G.E < E){
GRAPHinsertE(G, EDGE(RANDv(G), RANDv
(G)));G.E++;
}
return G;
}

```

对于给定的顶点数 $V$ ,通过生成 $0 \sim V-1$ 之间的整数对来生成随机边,自然它也存在平行边和自循环。随机图一个重要的数学模型就是以概率 $p$ 生成每条边。由于满图的边数是 $V(V-1)/2$ ,所以当选择概率 $p=2E/[V(V-1)]$ 时,可以得到边的期望数为 $E$ 的

图。

```

Graph GRAPHrand(int V, int E)
{ int i,j;
double p = 2.0*E/V/(V-1);
Graph G = GRAPHinit(V);
for (i = 0; i < V; i++)
for(j = 0; j < i; j++)
if(rand() < p*RAND_MAX)
GRAPHinsertE(G, EDGE(i, j));
return G;
}

```

由于没有检测边的总数是E,所以该算法每次生成的图的边数只是大体上为E,并不一定等于E,但是它排除了重复边和自循环。图2和图3是生成的两个随机图G(60,50)(稀疏图)和G(60,500)(稠密图)。

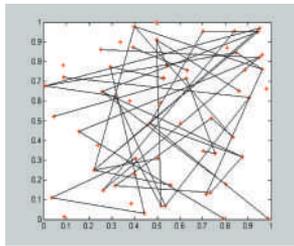


图2 G(60,50)

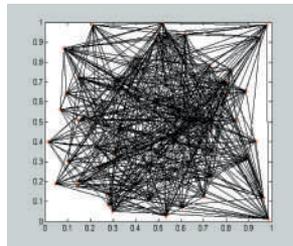


图3 (60,500)

上述算法给出的图的边数大体上等于E,如果需要图的边数为E。则需在上述算法的基础上进行改进。

```

int FINDe(Graph G, int v, int w)
{ link t = G.adj[v];
  if(v == w) return 0;
  else{
  if(t == NULL) return 1;
  while( t != NULL){ if(t->v == w) return 0; t = t->
next;
}
  return 1;
}
}

```

```

Graph GRAPHrandEreal(int V,int E)
{ int flag,v,w;
Graph G = GRAPHinit(V);
while(G.E < E){
v = RANDv(G); w = RANDv(G);
if(FINDe(G, v, w)) { GRAPHinsertE(G, EDGE(v,
w)); G.E++; }
}
}

```

```

return G;
}

```

首先增加对边数的检测,保证生成边的总数为E,同时增加一个函数,来查找是否有重复边和自循环,从而使生成的边的总数达到要求并且没有重复边和自循环。

以上几种算法生成的图可能并不满足实际研究对象的需要,如地图、电路、网络以及其它情况下的模型。因为这些类型的图还展示了图的位置性,即图的顶点在平面上的坐标,下面将为这类图建模。

### 2.2 K-邻接图

当随机挑选一顶点v,再从顶点编号为v的k倍内的顶点中挑选出次顶点w,连接v-w就得到K-邻接图,如果把它们安排成环,还可以体现出该类图的位置性。

```

int MIN(int dif, int V)
{ if (dif <= V/2) return dif; return V-dif; }
Graph GRAPHrandK(int V, int E, int K)
{ int i,j;
Graph G = GRAPHinit(V);
while(G.E <= E){
i = rand() % V;j = rand() % V;
if (MIN(abs(j - i), V) < K)
{ GRAPHinsertE(G, EDGE(i,j)); G.E++; }
}
return G;
}

```

算法中需要说明的是,由于顶点分布是圆形,所以与边相连的两顶点的编号差abs(j-i)取顺时针方向和反时针方向最小者MIN(abs(j - i), V)。图4是通过该算法得出的33个顶点和99条边,且限制每条边连通的编号差小于10的图。

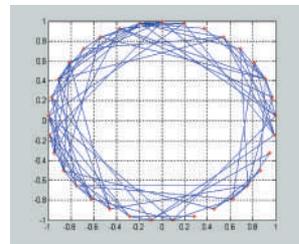


图4 K邻接图G(33,99)

### 2.3 欧几米德邻接图

首先在平面上生成V个随机点,然后连通距离小于阈值d的任何两顶点,如果d值较小则生成稀疏图,如果d值较大,则图是稠密的。这个图为处理地图、电路等类型的图建模,在这样的图中,顶点的

坐标是确定的。这种模型更容易使问题形象化,而且更能展示算法的性质。需要强调的是该模型生成的图对  $d$  非常敏感,当  $d$  相对于  $V$  的最大坐标值变化 1%,就会对边数有较大影响。

```

a=rand(1,V)
b=rand(1,V)
plot(a,b,'b. ');f=0.15;k=0;
for i=1:V
    for j=1:V
        if i~j
            s=((a(i)-a(j))^2+(b(i)-b(j))^2)^0.5;
            if s<f
                x=[a(i),a(j)];
                y=[b(i),b(j)];
                line(x,y,'Color','r','LineWidth',1);
                k=k+1;
            end
        end
    end
end
end

```

#### 注释及参考文献:

- [1]张红霞,白桂梅.数据结构与实训[M].北京:电子工业出版社,2008:148-149.
- [2]Robert Sedgewick . Algorithms In C[M]. by Robert Sedgewick,2002:36-38.
- [3]耿素云,屈婉玲,王捍贫.离散数学教程[M].北京:北京大学出版社,2004,2.

## The Research into Generating Algorithm of the Graph about Adjacent List

DONG Jia-qiang

(Xichang College, Xichang, Sichuan 615013)

**Abstract:** In this paper, the author introduces several kinds of important graphs and its generating algorithms in detail, including random graph,  $K$ -adjacent graph and Euclidean adjacent graph. For every kind of graph, the author uses C language to describe its formation and algorithms, and analyzes the functions of each algorithm in brief. Meanwhile, the author gets a specific graph after using a set of data to demonstrate every algorithm.

**Key words:** Adjacent list; Graph; Algorithms

end

end

这个算法是由 Matlab 来实现的。图 5、图 6 是  $V$  的坐标为  $(0-1)$ ,  $d$  分别为 0.05, 0.2 所得到的欧几米德邻接图,从图上证实了  $d$  对边数的影响。

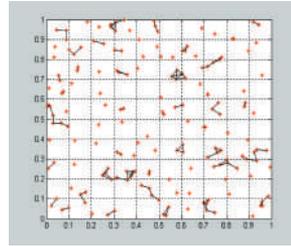


图 5  $d=0.05$

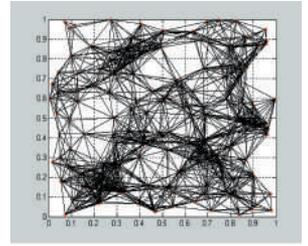


图 6  $d=0.2$

### 3 结束语

通过对上述几种类型的图及生成算法的探讨,有助于解决软件开发中许多与图相关的实际问题。图是一个复杂的组合对象,比一般的数据结构要复杂的多,实践表明,对图生成算法的探讨和实现有助于对图的性能进行理解和测试。