

最短路径算法的研究

陈 煜¹, 吴力合²

(1. 武警工程学院研究生队; 2. 武警工程学院通信工程系, 陕西 西安 710086)

【摘要】 分析了解决最短路径问题的几种算法的适用情况, 同时对这几种算法进行了比较, 有助于在处理实际问题时合理选择算法。

【关键词】 最短路径; 算法; 动态规划; A*; Dijkstra; Floyd

中图分类号: 024 文献标识码: A 文章编号: 1811-8755(2005)01512

1. 引言

求最短路径是一种十分常见的问题。如进行网络路由选择时要找一条从源节点到目标节点的最短路径即经过的时延最短, 地图上各城市间的最短路线问题, AOE网上的关键路径问题, 都可看作最短路径问题。以上这些问题都可抽象成对应的拓扑结构图, 图上的结点和边表示对应的含义, 要求解的问题就是结点之间的最短路径。

2. 算法分析

拓扑结构如图 1 所示, 求 A 到 D 之间的最短路径。

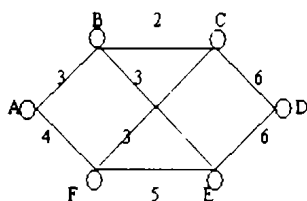


图 1

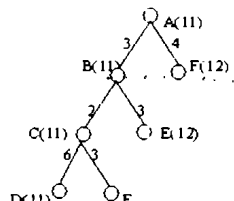


图 2

此类问题最基本的解法是求出 A 到 D 所有路径的长度, 取出其中最短的路径。但当拓扑结构比较复杂时, 该方法就失效, 而且难以用计算机求解, 因此一般不采纳。

算法 I: 动态规划法

此算法主要用于解决运筹学的动态规划问题, 对求解最短优路径同样奏效。

基本思想是: 将一个问题划分为若干步来完成, 可以证明整个问题的最优解必定与子问题的一致。

将问题求解过程划分为 3 步, 为 $A \rightarrow (B, F) \rightarrow (C, E) \rightarrow D$ 。用 X_k 表示 k 步上的结点, $F_k(X_k)$ 表示结点 X_k 到目标结点的最路径长, $Y(X_k)$ 表示结点 X_k 取得 $F_k(X_k)$ 时后继结点为 Y 。且有 $F_k(X_k) = \min\{|X_k X_{k+1}| + F_k(X_{k+1})\}$ 。

$X_k, F_k(X_k), X_{k+1}(X_k)$ 对应关系如表 1 所示:

X_3	C	E	X_2	B	F	X_1	A
$F_3(X_3)$	6	6	$F_2(x_2)$	8	9	$F_1(X_1)$	11
$Y(X_3)$	D	D	$Y(x_2)$	C	C	$Y(X_1)$	B

表 1

所以, A 到 D 的最短路径为 ABCD, 其长度为 11。

算法 II: A* 算法

此算法是搜索策略中的一种全局择优算法。

基本思想是: 给结点 x 一个估价函数 $f(x) = g(x) + h(x)$, 其中 $g(x)$ 为代价函数, $h(x)$ 为启发函数。选取 $f(x)$ 最小的 x 进行扩展搜索, 而启发函数 $h(x)$ 的选取是至关重要的, 选取 $h(x) \leq h^*(x)$, 其中 $h^*(x)$ 是结点 x 到目标结点的最小代价, 则该搜索策略是完备的。

可选取 $h(x)$ 为结点 x 到目标结点各段路径的最小值之和, 满足 $h(x) \leq h^*(x)$ 。

$$h(A) = 2 + 3 + 6 = 11, f(A) = 0 + 11 = 11.$$

1. 将 A 扩展为 B, F。

$$h(B) = 2 + 6 = 8, f(B) = 3 + 8 = 11; h(F) = 2 + 6 = 8, f(F) = 4 + 8 = 12. f(B) < f(F)$$

2. 将 B 扩展为 C, E。

$$h(C) = 6, f(C) = 5 + 6 = 11; h(E) = 6, f(E) = 6 + 6 = 12. f(C) < f(E)$$

3. 将 C 扩展为 D, F, 到目标结点 D. $f(D) = 11$ 。

搜索过程如图 2 所示(括号内的数表示估价函数值)。

所以, 最短路径为 ABCD, 其长度为 11。

算法 III: Dijkstra 算法

该算法的基本思想是: 设置顶点集合 S 并不断扩充这个集合, 一个顶点属于 S 当且仅当源到该顶点的最短路径已知。初始时, S 中仅含有源结点。设 V 是 G 的某一顶点, 把源到 V 且中间经过 S 中顶点的~~路径~~称为特殊路径, 并用数组 D 来记录当前每个顶点的最特殊路径。每次从 $V-S$ 中取出具有最特殊路径的顶点 V , 并将 V 添加到 S 中, 同时对 D 作必要的修改。一旦 S 包含了所有的结点, D 就记录了所有其他结点的最短路径长度。在计算过程中作相应的记录, 可得最短路径。

求解过程如表 2 所示:

步数	W	S	D[B]	D[C]	D[D]	D[E]	D[F]
1		{A}	3(A)	∞	∞	∞	4(A)
2	B	{A, B}		5(B)	∞	6(B)	4(A)
3	F	{A, B, F}		5(B)	∞	6(B)	
4	C	{A, B, F, C}			11(C)	6(B)	
5	E	{A, B, F, C, E}			11(C)		
6	D	{A, B, F, C, E, D}	3(AB)	5(ABC)	11(ABCD)	6(ABE)	4(AF)

表 2

所以, 最短路径为 ABCD, 其长度为 11.

算法 IV: Floyd

该算法可求出所有顶点间的最短路径.

其基本思想是: 设 $V = \{1, 2, \dots, n\}$, $G = \{V, E\}$, 设一个 $n \times n$ 矩阵 A, 初始时 $A(i, j) = \{0 \mid i=j; C[i, j] \mid i \neq j \text{ 且 } (i, j) \in E; \infty \mid i \neq j \text{ 且 } (i, j) \notin E\}$, 在矩阵 A 上做 n 次迭代. 经过 k 次迭代之后, A[i, j] 上的值是从顶点 i 到 j, 且中间经过编号大于 k 的最短路径长.

迭代公式为: $A^k[i, j] = \min \{A^{k-1}[i, j], A^{k-1}[i, k] + A^{k-1}[k, j]\}$.

这样 n 次迭代后, $A^n[i, j]$ 包含所有的最路径长.

为了确定相应的最短路径, 设置一个 $n \times n$ 矩阵 P, 当 k 使 A[i, j] 达到最小时, 置 $P[i, j] = k$, 约定 $P[i, j] = 0$ 表示顶点 i 到 j 的最短路径就是从 i 到 j 的边, 则根据矩阵 P 就可得到相应的最短路径.

模型的求解程序如下(在 matlab 环境下运行, {1, 2, 3, 4, 5, 6} 对应为 {A, B, C, D, E, F}):

```
function [A, P]=path(m, n)
A=[0, 3, inf, inf, inf, 4; 3, 0, 2, inf, 3, inf; inf, 2, 0, 6, inf, 3;
inf, inf, 6, 0, 6, inf; inf, 3, inf, 6, 0, 5; 4, inf, 3, inf, 5, 0];
for i=1:6 for j=1:6 P(i, j)=0; end end
for k=1:6 for i=1:6 for j=1:6 if A(i, k)+A(k, j)<A(i, j)
A(i, j)=A(i, k)+A(k, j); P(i, j)=k; end end
end end
k=P(m, n); if k>0
path(m, k); k path(k, n); end
运行该程序得(在命令窗口中输入 [A, P]=path(1, 4)):
k = 2, k = 3
A=[0, 3, 5, 11, 6, 4; 3, 0, 2, 8, 3, 5; 5, 2, 0, 6, 5, 3; 11, 8, 6, 0, 6, 9; 6, 3, 5, 6, 0, 5; 4, 5, 3, 9, 5, 0]
P=[0, 0, 2, 3, 2, 0; 0, 0, 0, 3, 0, 3; 2, 0, 0, 0, 2, 0; 3, 0, 0, 0, 3, 2; 0, 2, 0, 0, 0, 0; 0, 3, 0, 3, 0, 0]
```

由矩阵 a 可得到所有结点之间的最短路径长, 其中

A 到 D 之间的最短路径长为 $a(1, 4) = 11$, $k=2, k=3$, 说明最短路径经过 B, C, 即最短路径为 ABCD.

3. 算法比较

以下几个方面对以上算法进行比较:

(1) 适用对象: 算法 I 主要适用于有向图, 在要求不高时也可用于无向图; 算法 II 适用于无向图; 算法 III, IV 对于两种图均适用.

(2) 局限性: 算法 I 对步骤分明的问题易于求解(如图 1), 而对步骤不好划分的图(如图 3, 4)很难求解; 算法 II 对路径步数一样的易求解(如图 1, 4), 而对步数不等的(如图 3)难求解; 算法 III, IV 无此局限性.

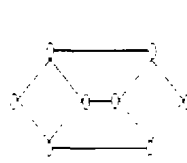


图 3

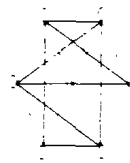


图 4

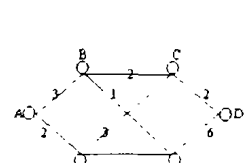


图 5

(3) 编程实现: 算法 I 由于要划分各个步骤, 计算机难以做到, 因此难以编程实现; 算法 II, III, IV 相对而言易于实现.

(4) 信息含量: 算法 I 解有向图求解时, 包含所有结点到目标结点的最短路径信息, 而解无向图求解时, 只有源结点到目标结点的最短路径可靠, 其他不可靠(如图 5 中点 E, 算法得到的信息为 6, 而最短路径长为 5: EBCD); 算法 II 只包含源结点到目标结点的最短路径信息; 算法 III 包含源结点到所有结点的最短路径信息; 算法 IV 包含所有结点之间的最短路径信息.

(5) 算法复杂度: 对于易于编程实现的算法 II, III, IV, 其时间复杂度(最坏情况下)分别为 $O(n^2)$, $O(n^2)$, $O(n^3)$, 可见算法的信息含量是以牺牲复杂度为代价的.

4. 结束语

解决最短路径问题的算法有动态规划法, A* 算法, Dijkstra 算法, Floyd 算法等等. 这些算法在适用对象, 局限性, 编程实现, 信息含量及算法复杂度方面有很大区别. 所以在处理实际问题时要综合考虑各方面的因素, 慎重选择算法.

参考文献:

[1] 运筹学适用教材组编. 运筹学. 清华大学出版社, 1985
 [2] 王永庆. 人工智能原理与方法. 西安交通大学出版社, 1998
 [3] 傅清祥, 王晓东编著. 算法与数据结构(第二版). 电子工业出版社, 2001

的各项基本设施较完善且费用较低，具有外部经济；到了发展中期，这种情形会逐渐地趋于缓慢；到了发展后期，因核心区出现外部不经济的现象，地价逐渐高涨、交通阻塞、噪音、空气污染及其它问题，人们便渐渐地转向外围区域发展，区域不均衡因此缩小。

当然，区域经济非均衡增长理论还有其他模型，这里我们只对影响较大的理论做一回顾。另外，各个理论分界也不是像以上描述的那样清晰，更多的是为解释同一个经济现象；或者即使是一脉相承的理论，也在研究方法和关注点上截然不同。所以说，对理论的分类和归纳方式也是灵活多样的，其目的只是让有兴趣关注区域经济发展的学人对非均衡增长理论有个整体、清楚的了解。

【参考书目】

1. 郝寿义，安虎森. 《区域经济学》. 经济科学出版社，2004年版.
2. 周起业等. 《区域经济学》. 中国人民大学出版社，1989年版.
3. 赫希曼. 《. 经济发展战略》. 经济科学出版社，1991年版.
4. 宋栋. 《中国区域经济转型发展的实证研究——以珠江三角洲为例》. 经济科学出版社，2000年版.

上接 107 页

时，强化企业会计报表审计制度，注册会计师应对企业会计报表的合法性、恰当性和一致性进行独立、客观、公正的审查验证，切实提高会计报表质量。

中小私营企业财务管理从总体上看，仍处于低效率低水平阶段。财务管理与私营企业日益发展壮大的要求不相适应。要使我国私营企业财务管理为企业的稳定增长、长远发展和使企业具有强壮生命力和优越的竞争力作出更大贡献，就应进一步优化企业理财环境，突出财务管理的战略地位，健全财务管理机构，并将其放在高层管理的位置上，赋予财务管理部门更大的权力和责任。要按市场经

上接 73 页

作者简介：

陈煜（1981-），男，湖北天门人，学员，武警工程学院硕士研究生，主要研究方向为神经网络，决策支持系统。

5. 张敦富，覃成林. 《中国区域经济差异与协调发展》. 中国轻工业出版社，2001年版.
6. 韦伟. 《中国经济发展中的区域差异与区域协调》. 安徽人民出版社，1995年版.
7. 张燕. “西方区域经济理论综述”. 中国人民大学复印报刊资料. 2004年第4期.
8. 安虎森. “区域经济非均衡增长与区域空间二元结构的形成”. 延边大学社会科学报，1997年第1期.
9. 安虎森：“增长极理论评述”. 南开经济研究，1997年第1期.
10. 徐滇庆：“不平衡发展是区域经济的一般规律”. 经济信息报，1997年5月30日.

作者简介：

汪茂泰（1976-），安徽安庆人，现就读于云南民族大学经管学院，区域经济学专业；

王永志（1974-），湖北黄冈人，现就读于云南民族大学人文学院，专门史专业。

（收稿日期：2005-1-13）

济的要求，进一步转变观念更新内容、转变职能，建立健全财务管理人员的激励机制，充分调动管理人员理财的积极性、创造性、采取有效措施切实提高私营企业财务管理水平。作者

作者简介：

林八均（1968.12-），男，学历：大学本科，职称：会计师，中国注册会计师，中国注册资产评估师，注册税务师，注册房地产估价师，注册土地估价师，现任永安燕江有限责任会计师事务所主任会计师（所长），永安市政协常委，永安市工商联常委等职务，曾在国家级和省级发表了多篇论文。

（收稿日期：2005-2-6）

（收稿日期：2005-1-21）