

用元胞自动机求最短路径的一种新算法

孙 强¹, 戴志军²

(1. 华东师范大学 计算机科学技术系, 上海 20024;
2. 华东师范大学 河口海岸科学研究院, 上海 200241)

摘 要: 在一个基于元胞自动机模型的求图中一个顶点到另一个顶点的最短路径的算法的基础上, 分析出: 它的关键部分(即演化规则)中“减最小剩余权”这一重要步骤与求图中一个顶点到另一个顶点的经典的最短路径算法的基本思想相距甚远, 应该省去。在提出的新算法中省去“减最小剩余权”这一重要步骤, 这个改进较大地提高了算法的效率。最后通过举例子分别用两个算法进行求解, 通过这些求解步骤的对比, 明显看出本算法的正确性和高效性。

关键词: 元胞自动机; 最短路径算法; 减最小剩余权; 省略

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)02-0042-03

A New Shortest Path Algorithm Using Cellular Automata Model

SUN Qiang¹, DAI Zhi-jun²

(1. Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China;
2. Institute of Estuary and Coastal Research, East China Normal University, Shanghai 200241, China)

Abstract: On the base of one shortest path algorithm from one point to another point in a graph based on cellular automata model, it is pointed out that its critical step of “subtraction of the smallest surplus weight” in evolution rule of the algorithm is far from the basic thinking of classic shortest path algorithms from one point to another point in a graph, it should be left out. The critical step of “subtraction of the smallest surplus weight” is left out in a new algorithm of this article to raise the efficiency of algorithm greatly. At last, one example is used to show correctness and efficiency of this new algorithm by comparing evolution steps of these two algorithms.

Key words: cellular automata; shortest path algorithm; subtraction of the smallest surplus weight; leave out

0 引 言

求图的最短路径是多年来一个十分引人注目的研究方向^[1~6], 它在地理学^[3]、自动控制、运筹学、计算机科学、交通、军事等很多方面有着很重要的应用。元胞自动机(Cellular Automata)是一种时间、空间、状态都离散的动力学模型, 是非线性科学的一种重要的研究方向, 它适合于复杂系统时空演化过程的动态模拟研究。2004年5月, 吴晓军、薛惠锋提出了一个“基于元胞自动机扩展模型的图的最短路径算法”^[2](以下简称此文为吴文); 在吴文的基础上, 2006年5月, 李慜等提出了一种“改进的基于元胞自动机扩展模型的图的最短路径算法”^[1](简称李算法, 以下简称此文为李文), 李算法用元胞自动机求图中一个顶点到另一个顶点的最短路径, 确实有新意; 文中在李算法的基础上作

了改进, 分析出: 它的关键部分(即演化规则)中“减最小剩余权”的这一重要步骤与求最短路径的经典 Dijkstra 算法的基本思想相距甚远, 应该省去; 文中提出的改进算法中省去了“减最小剩余权”这一步骤, 这个改进较大地提高了算法的效率。在理论分析的基础上, 最后通过举例子分别用李算法和文中提出的算法进行求解, 通过这些求解步骤的对比, 明显看出本算法的优越性。

1 最短路径

图的最短路径的研究主要包括以下四个方面:

- (1) 求图中从一个顶点到其余顶点的最短路径, 如 Dijkstra 算法;
- (2) 求图中任意二个顶点的最短路径, 如 Floyd 算法;
- (3) 求图中带受限制条件(如: 顶点数受限制, 或带单一限制条件)的最短路径^[5]。
- (4) 求图中一个顶点到另一个顶点的所有最短路

收稿日期: 2008-06-29

基金项目: 国家自然科学基金资助(40771200)

作者简介: 孙 强(1963-), 男, 安徽合肥人, 副教授, 硕士, 主要从事软件与算法研究; 戴志军, 副教授, 硕士, 主要从事河口海岸研究。

径^[6]。

2 元胞自动机

一般元胞自动机^[1,2]通常包括四个要素:元胞空间、元胞状态、邻居和转换规则。用数学符号可以将一个标准元胞自动机表示为一个四元组 $A: A = (L_d, S, N, f)$ 。这里 A 表示一个元胞自动机系统; L_d 表示 d 维元胞空间, d 是个整数, 表示元胞自动机内元胞空间的维数; S 是元胞内有限的、离散的元胞状态集合; N 表示一个所有邻域内元胞的组合(包括中心元胞); f 是基于邻近元胞的演化规则。元胞自动机中所有的元胞都遵循相同的规则演化, 而且, 某一时刻的一个元胞状态只与前一时刻此元胞的状态以及前一时刻邻居元胞的状态有关。

3 李算法的演化规则

李算法的元胞演化规则如下:

元胞的演化状态分为 4 个状态, 即: 空(S_N)、生长(S_G)、繁殖(S_B)、成熟(S_M)。若中心元胞状态为

(1) S_M 状态, 表示该元胞处于成熟状态, 已被寻路, 不做变化;

(2) S_B 状态, 表示此顶点已经被寻路, 下一时刻翻转到 S_M ;

(3) S_N 状态, 表明该元胞处于空闲状态, 检查前一时刻邻居的元胞状态;

若存在任一邻居 j 为 S_B 状态, 则中心元胞 i 翻转到 S_G , 修改中心元胞 i 的剩余权为

$$R_i = \text{distance}(i, j), i, j \in L_d$$

其中 $\text{distance}(i, j)$ 为元胞 i 到元胞 j 之间的距离。

(4) S_G 状态, 表明该元胞处于生长状态, 检查前一时刻邻居元胞状态, 定义元胞空间上的最小剩余权为 mr , 令 $R_i = R_i - mr$, 若 $R_i = 0$, 则中心元胞 i 在下一时刻状态翻转为 S_B 。否则, 若中心元胞 i 存在任一邻居元胞 j 的状态为 S_B , 且 $\text{distance}(i, j) < R_i$, 则更新 R_i 为: $R_i = \text{distance}(i, j)$, $i, j \in L_d$, 同时调整下一时刻最小剩余权 mr' 为: $mr' = \min\{R_i, i \in L_d\}$ 。

李文^[1]中举了一个例子演示了元胞转换过程, 但此例有点简单了, 文中举了一个例子如图 1 所示, 然后按照李算法画出求图 1 从 v_0 到 v_5 最短路径的过程中元胞演化表的每个步骤(见表 1)。

在表 1 中, 按照李算法共进行了 6 次“减最小剩余权”操作。(含: 从演化步骤 1 向演化步骤 2 演化过程中 1 次, 从步骤 3 向步骤 4 演化过程中 2 次, 从步骤 5

向步骤 6 演化过程中 1 次, 从步骤 7 向步骤 8 演化过程中 2 次。)

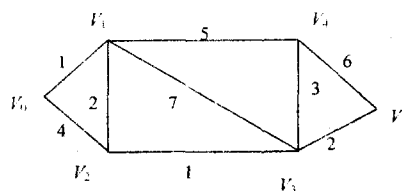


图 1 一个无向图

表 1 用李算法求图 1 最短路径的元胞演化表

演化	v_0	v_1	v_2	v_3	v_4	v_5	$\min(r)$
0	S_B	S_N	S_N	S_N	S_N	S_N	0
1	S_M	$S_G/1$	$S_G/4$	S_N	S_N	S_N	1
2	S_M	S_B	$S_G/3$	S_N	S_N	S_N	0
3	S_M	S_M	$S_G/2$	$S_G/7$	$S_G/5$	S_N	2
4	S_M	S_M	S_B	$S_G/5$	$S_G/3$	S_N	0
5	S_M	S_M	S_M	$S_G/1$	$S_G/3$	S_N	1
6	S_M	S_M	S_M	S_B	$S_G/2$	S_N	0
7	S_M	S_M	S_M	S_M	$S_G/2$	$S_G/2$	2
8	S_M	S_M	S_M	S_M	S_B	S_B	0

下面以此图 1 为例, 来说明文中算法的改进。

4 文中改进: 省去“减最小剩余权”步骤

4.1 省去“减最小剩余权”的原因

李算法的关键是其演化规则, 其中第(4)步又是演化规则中的关键; 在第(4)步中要求出同一行中所有处于生长状态 S_G 的元胞的最小剩余权 mr , 并用减法求出每个处于生长状态的元胞新的剩余权, 即: 令 $R_i = R_i - mr$, 文中认为: 进行 $R_i = R_i - mr$ 的减法操作没必要, 这一步骤可省去。因为用元胞自动机求最路径毕竟是一种模仿经典的 Dijkstra 算法的近似算法, 而 Dijkstra 算法的基本思想是: 要在求解过程中进行从源点到终点路径上各边长的逐步“累加”求和和累加后的比较, 而李算法中“减最小剩余权”($R_i = R_i - mr$)是做图中二个边长的减法, 这一“做减法”的操作与 Dijkstra 算法的“累加”基本思想相距甚远; 另外, 用 Dijkstra 算法求图 1 中从 v_0 到 v_5 的一条最短路径所进行的操作一定是在从 v_0 到 v_5 的无回路的路径上各边长的累加操作, 而在李算法上述减法($R_i = R_i - mr$)所表示的实质上是 2 条边长的相减操作中, 这 2 条边也不一定在从源点到终点的一条无回路的路径上(如: 表 1 中从演化步骤 1 向演化步骤 2 的演化过程中做的一次减法($S_G/4$ 中的“4”减最小剩余权值“1”)实质上是边(v_0, v_2)的长度“4”减去边(v_0, v_1)的长度“1”, 这 2 条边不可能在从 v_0 到 v_5 的一条无回路的路径上。), 所以做“减最小剩余权”减法的结果无多大的意义。

4.2 文中的演化规则

文中演化规则的(1)、(2)、(3)步与李算法一样,第(4)步省去了“减最小剩余权”步骤,文中元胞演化规则如下:

若中心元胞状态为

(1) S_M 状态,表示该元胞处于成熟状态,已被寻路,不做变化;

(2) S_B 状态,表示此顶点已经被寻路,下一时刻翻转到 S_M ;

(3) S_N 状态,表明该元胞处于空闲状态,检查前一时刻邻居的元胞状态;

若存在任一邻居 j 为 S_B 状态,则中心元胞 i 翻转到 S_G ,修改中心元胞 i 的剩余权为

$$R_i = \text{distance}(i, j)$$

其中 $\text{distance}(i, j)$ 为图中顶点 i 到顶点 j 之间的边长。

(4) S_G 状态,表明该元胞处于生长状态,如同一行中无元胞处于 S_B 状态,则找出同一行中处于生长状态的元胞的最小剩余权值为 mr 和具有此剩余权的元胞 i ,元胞 i 的状态下一时刻翻转为 S_B ;如同一行中有元胞处于 S_B 状态,且如元胞 k 存在任一邻居 j 的状态为 S_B ,且 $\text{distance}(k, j) < R_k$,则更新 R_k 为: $R_k = \text{distance}(k, j)$ 。

相应的元胞状态转换图如图 2 所示。

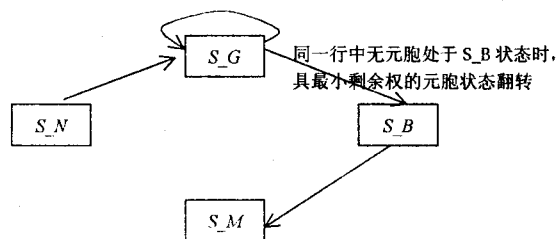


图 2 元胞状态转换图

用文中算法求图 1 的从 v_0 到 v_5 的最短路径的元胞演化状态表见表 2。

所求得的最短路径 v_0, v_1, v_2, v_3, v_5 与用李算法以及用 Dijkstra 算法求解结果完全一致,而且表 2 与表 1 相比共省去 6 次“减最小剩余权”的操作,效率的提高较明显。

表 2 文中的算法的演化状态表

演化	v_0	v_1	v_2	v_3	v_4	v_5
0	S_B	S_N	S_N	S_N	S_N	S_N
1	S_M	$S_G/1$	$S_G/4$	S_N	S_N	S_N
2	S_M	S_B	$S_G/4$	S_N	S_N	S_N
3	S_M	S_M	$S_G/2$	$S_G/7$	$S_G/5$	S_N
4	S_M	S_M	S_B	$S_G/7$	$S_G/5$	S_N
5	S_M	S_M	S_M	$S_G/1$	$S_G/5$	S_N
6	S_M	S_M	S_M	S_B	$SCG/5$	S_N
7	S_M	S_M	S_M	S_M	$S_G/3$	$S_G/2$
8	S_M	S_M	S_M	S_M	$S_G/3$	S_B

5 结束语

在李算法基础上,文中加以改进;首先在理论上分析出它的演化规则中“减最小剩余权”这一重要步骤与求最短路径的成熟算法基本思想相距甚远;然后在自己的算法中将这一步骤省去,并用一个实例的 2 种求解过程的对比来说明文中算法的有效性;因为文中提出的算法可省去多次减法操作,所以与李算法相比,文中提出的算法效率有明显的提高。我们的工作是对元胞自动机求最短路径这一方法的新的探索。

参考文献:

- [1] 李 慧,薛惠锋,吴晓军.改进的基于元胞自动机扩展模型的图的最短路径算法[J].微计算机应用,2006,27(3):365-367.
- [2] 吴晓军,薛惠锋.基于元胞自动机扩展模型的图的最短路径算法[J].计算机应用,2004,24(5):92-93.
- [3] Zhan F B. Three Fastest Shortest Path Algorithms on Real Road Networks: Data structures and Procedures[J]. Journal of Geographic Information and Decision Analysis,1997(1):69-82.
- [4] 孙 强,沈建华,顾君忠.Dijkstra 的一种改进算法[J].计算机工程与应用,2002(3):99-101.
- [5] 孙 强,杨宗源.求受顶点数限制的最短路径问题的一个算法[J].计算机工程,2002,28(9):73-74.
- [6] 孙 强,沈建华,顾君忠.求图中顶点之间所有最短路径的一种实用算法[J].计算机工程,2002,28(2):134-136.

(上接第 41 页)

- [3] 边肇祺,张学工.模式识别[M].第 2 版.北京:清华大学出版社,2000.
- [4] 王国胤,Rough 集理论与知识获取[M].西安:西安交通大学出版社,1999.
- [5] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science, 1982,11(5):341-356.
- [6] 周 瑛,张 铃.模糊集方法在检索评价系统中的应用

- [J].计算机技术与发展,2007,17(1):111-113.
- [7] 谭 优,王泽勇.图像阈值分割算法实用技术与比较[J].微计算机信息,2007(6):298-299.
- [8] Li C J. Fuzzy homogeneity and scale-space approach to color image segmentation[J]. Pattern Recognition,2003,36:1545-1562.