一种基于 GIS 最短路径搜索的 A* 改进算法

An Improved A* Algorithm in the Shortest Path Searching Based on GIS

王　肖　徐友春　章永进　郭振东　（解放军军事交通学院 汽车工程系 天津 300161）

摘　要： 目前在 GIS 领域，最短路问题是研究和应用的重点，其中最短路径搜索算法的效率问题是普遍关注和在实际应用中迫切需要解决的问题. 本文分析目前几种流行的最短路径算法并指出它们的优缺点，提出了一种利用蚁群算法和遗传算法进行改进的 A* 算法，并对其进行了必要的推导.。实验证明，改进的 A* 算法在 VC + + 6.0 和 MapX 控件环境下具有较好的可行性和适用性。

关键词： 地理信息系统 最短路径算法 遗传算法 蚁群算法 改进 A* 算法

近些年来，随着计算机的普及以及地理信息科学的发展，地理信息系统（GIS）在交通、军事、土地资源管理、城市规划等方面都得到了广泛且深入的应用. 这些应用领域中的地理信息系统经常涉及最短路径搜索问题。最短路径不仅仅指一般地理意义上的距离最短，还可以引申为最低费用、最快时间问题等。其实无论是距离最短、时间最快还是费用最低，它们的核心都是最短路径算法。

目前对基于地理信息系统的最短路径搜索算法的研究很多，其中主要研究算法有 Dijkstra 算法、Bellman – Ford – Moore 算法、Floyd 算法、A* 启发式算法等、B* 启发式算法等。论文综合分析了 Dijkstra 算法、A* 启发式算法等、B* 启发式算法的优缺点，选定 A* 启发式算法作为研究的算法，并采用蚁群算法和遗传算法相融合的方式对其进行改进。

1　几种算法的比较分析

Dijkstra 算法是 E. W. Dijkstra 于 1959 提出来的。其适用范围为所有弧的权均为非负的道路网络。它是目前公认的求解最短路问题高效的经典算法之一，时间复杂度为 $o(n^2)$，其中 n 为结点数。其基本思想是从起点 x 出发，逐步地向外探寻最短路。执行过程中，与每个点对应，记录下一个数（称为这个点的标号），它或者表示从 x 到该点的最短路的权（称为 p 标号），或者是从 x 到该点的最短路的权的上界（称为 T 标号），方法的每一步是去修改 T 标号，并且把某一个具 T 标号的点改变为具 p 标号的点，从而使 N 中具 p 标号的顶点数多一个，这样，至多遍历整个结点集 N，就可以求出从 x 到各点的最短路。Dijkstra 算法的搜索是盲目式的，其优点为不需要涉及具体问题的相关信息，只需要结点和路段之间的连接关系即可进行搜索，其解是全局最优的。当然其缺点也很明显，由于其搜索是盲目式的，所以搜索的范围大，时间长。Dijkstra 算法提出以后，有很多学者对其具体实施方法进行了改进，使其求解速度有所提高。

A* 算法是由 Hart、Nilsson、Raphael 等人首先提出的，它实际是一种启发式搜索。不同于 Dijkstra 算法的盲目式搜索，也不同于广度优先搜索，该算法在搜索过程中加入与问题域相关的启发因子来缩小搜索范围，加快搜索速度。具体到最短路问题，就是在选择下一个被检查的结点时，除了考虑已知的局部信息（起点到该结点的距离），还对当前结点距终点的距离做出估计，作为评价该结点处于最优路线上的可能性的量度，这样就引入了全局信息，其搜索不在是盲目式的，搜索范围大大缩小。图 1 为 Dijkstra 算法与 A* 启发式算法算法的搜索范围的比较。由图 1 可见，Dijkstra 算法的搜索是以起点为圆心以圆的方式向外拓展，而 A* 算法则大体上朝着终点方法搜索，其效率要优于 Dijkstra 算法。

B* 算法其实质是一种改进的 A* 算法。A* 算法与 Dijkstra 算法的最大差异就是在搜索过程中引入估价函数（与问题域相关的启发因子）。估价函数的优劣决定了 A* 算法的运算效率。B* 算法采用一个新估价函数代替 A* 算法的估价函数。其具体实现是对当前

结点距终点的距离作估计时,估计该结点的所有子结点距终点的距离,然后加上该结点到它的实际距离,取其最小值。就估价函数的值来说,B* 算法的确优于 A* 算法,但其估价函数的计算时间要远大于 A* 算法,而且其估价函数的计算是以穷举该结点的所有子结点为代价的,所以相对 A* 算法来说,其搜索不但没有缩小,而且还有所扩大,其搜索时间也较长。因此本论文采用 A* 启发式算法作为搜索算法。

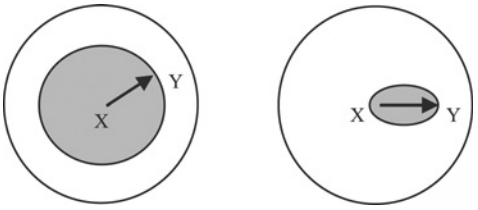

图 1　Dijkstra 和 A* 启发式算法算法搜索范围比较

2　改进的 A* 算法在最短路径搜索中的高效实现

A* 算法是到目前为止最快的一种计算最短路径的算法,但它只是一种‘ 较优 ’算法,即它一般只能找到较优解,而非最优解。

A* 算法成功与否的关键在于估价函数的正确选择,从理论上说,一个完全正确的估价函数是可以非常迅速地得到问题的正确解答,但一般完全正确的估价函数是得不到的,因而 A* 算法不能保证它每次都得到正确解答。一个不理想的估价函数可能会使它工作得很慢,甚至会给出错误的解答。经过蚁群算法和遗传算法改进后的 A* 算法能够迅速找到较正确估价函数,从而实现高效搜索到最短路。

2.1 遗传算法和蚁群算法介绍

遗传算法(GA)是美国 Michigan 大学 J. Holland 教授于 1975 年提出的。它是以达尔文生物进化理论“ 适者生存,优胜劣汰 ”和孟德尔遗传变异理论‘ 生物遗传进化主要在染色体上,子代是父代遗传基因在染色体上的有序排列 ”为基础,模拟生物进化过程。目前,遗传算法已广泛应用于自适应控制、组合优化、模式识别、机器学习、规划策略、信息处理等领域。

遗传算法的几个重要概念:

(1) 个体(Individual)。个体也称为染色体(Chromosome),对应了问题的一个解。

(2) 群体(Population)。个体的集合,一个群体包含问题一些解的集合。

(3) 适应度。适应度是遗传算法中衡量群体中各个体(问题的一个解) 优劣的尺度。根据适应度的大小,决定该个体是繁殖还是被淘汰。

蚁群算法是由意大利学者 M. Dorigo 等人于 1991 提出的。这种算法有别于传统编程模式,其优势在于,避免了冗长的编程和筹划,程序本身是基于一定规则的随机运行来寻找最佳配置。也就是说,当程序最开始找到目标的时候,路径几乎不可能是最优的,甚至可能是包含了无数错误的选择而极度冗长的。但是,程序可以通过蚂蚁寻找食物的时候的信息素原理,不断地去修正原来的路线,使整个路线越来越短,也就是说,程序执行的时间越长,所获得的路径就越可能接近最优路径。蚁群算法是一种新型的模拟进化算法,具有很好的通用性和鲁棒性,在解决组合优化问题方面有良好效果,但存在如计算时间较长、容易陷入局部最优等问题。

为模拟实际蚂蚁的行为,首先引入如下记号:

m——蚁群中蚂蚁的数量;

$b_i(t)$ ——t 时刻位于中间点的蚂蚁的个数,$m = \sum_{i=1}^{n} b_i(t)$;

d_{ij}——两中间点 i 和 j 之间的距离;

η_{ij}——边的能见度,反映由中间点 i 转移到中间点 j 的启发程度,这个量在蚂蚁系统的运行中不改变;

τ_{ij}——边(i,j) 上的信息素轨迹强度;

$\triangle\tau_{ij}^{k}$——蚂蚁 k 在边(i,j) 上留下的单位长度轨迹信息素量;

p_{ij}^{k}——蚂蚁 k 的转移概率,j 是尚未访问的中间点。

每只蚂蚁都具有如下特征:

(1) 在从中间点 i 到中间点 j 的运动过程中,蚂蚁在边(i,j) 上释放一种物质,称为信息素轨迹;

(2) 蚂蚁概率地选择下一个将要访问的中间点,这个概率是两中间点距离和连接两中间点的路径上存有轨迹量的函数;

(3) 为了满足问题的约束条件,在一次循环中不允许蚂蚁选择已经访问过的中间点。

2.2 融合蚁群算法和遗传算法的 A* 算法实现

如表 1 所示，首先建立网络分析和算法的数据结构：

1. 结点：每一结点都有一结点号作为其标识。其中经度与纬度标志其位置信息，父结点表示其在最优路径中的上一结点号，子结点数与该结点直接相连的结点数目，在 Power 表（路权表）中位置表示在 Power 表记录所有经过该结点的弧权起始记录位置，g、h、f 标志搜索过程该结点的优先级。

2. 结点表：路网结点的集合构成结点表，结点在表中的位置与结点号对应。这样要访问某一结点只需根据其结点号在结点中相应的位置就可以找到它。

表 1　数据结构

节点	道路的权	Open（Close）表	遗传/蚁群
节点号 经纬度	道路长度	节点指针	最小迭代次数 n1
父节点号 子节点号	起节点号	前一对象指针	最大迭代次数 n2
Power 表中位置 距起点实际距离 g	终节点号	后一对象指针	最小进化率 m
启发值 h 估价值 f			蚁群权值 k

3. 路权表：路权表记录路网中所有弧的权。路权表大体上按结点号排列，经过一结点的所有弧记录在一起，其起始位置包含在结点结构中。

4. Open（Close）表：设立 Open 表和 Close 表。Open 表用来存放将要处理的结点，Close 表用来存放已处理完毕的结点。Open 表与 Close 表均是双向链表。

5. 遗传/蚁群表：存放遗传算法的最小迭代次数 n1，最大迭代次数 n2，设置的最小进化率 m 和求蚁群初始信息素的权值 k。

利用遗传算法和蚁群算法寻找估价函数，先对路网中每个结点关联一个指引搜索方向的估价函数，估计从起点途经此结点到达终点所需的最小通行代价，搜索过程就是不断去拓展估价函数值最小的那个结点，直至到达终点。

估价函数由两部分组成：

$$f(n)=g(n)+h(n)$$

式中 $f(n)$ 为结点 n 估价函数，$g(n)$ 为从初始结点到结点 n 的实际代价，$h(n)$ 是从结点 n 到目标结点最佳路径的估计代价，称为启发函数。启发函数的选择十分重要，如果其值过小，启发函数就起不到启发的作用，导致搜索范围扩大（极端情况下，当 $h(n)=0$ 时，等同于 Dijkstra 算法）；如果其值过大，则不保证找到最优解，此时只能称为 A 算法。设从结点 n 到目标结点的最小代价为 $\hat{h}(n)$，可以证明当 $h(n)<\hat{h}(n)$ 时，能找到最优解。可以想象，如果 $h(n)=\hat{h}(n)$，那么算法将沿着最优路径搜索到终点。本论文中启发函数 $h(n)$ 取结点到目标结点直线距离。

在利用遗传算法和蚁群算法寻找估价函数的初期，各个个体的适应度比较离散，进化速率比较快，但到了后期，随着群体平均适应度与最大适应度的接近，其搜索效率显著降低。而蚁群算法在探索的初期由于缺乏信息素，搜索速度比较缓慢，但当信息素积累到一定的强度之后，向最优解收敛的速度迅速提高，如图 2 所示。因此可以考虑先用遗传算法搜索估价函数，等其搜索效率显著降低时，算法终止，并将其搜索结果生成蚁群算法的初始信息素的分布，然后转入蚁群算法求最优估价函数。

遗传算法的终止可设定如下终止条件：

（1）设置最小遗传迭代次数 n1 和最大遗传迭代次数 n2。

（2）在迭代过程中统计子代群体的进化率，并以此设置子代群体最小进化率 m，在设定的迭代次数范围内，如果连续 5 代子代群体的进化率都小于最小进化率，则可终止遗传算法。

蚁群算法初始信息素的生成：对遗传算法求得的解设定权值 k，其解的加权和可用来生成蚁群算法的初始信息素。

算法流程为：开始时将起点放入 Open 表，然后处理 Open 表中的结点，直到 Open 表空。Open 表中的结点是按估价值 f 的升序排列的。考察 Open 表中的第一个结点（f 值最小），对其所有子结点做出判断：如果不在 Open 表中，加入 Open 表；如果在 Open 表中，利用遗传算法和蚁群算法经过重新计算 g 值，重排 Open 表，找出最优估价值。当所有子结点处理完毕后，将该结点插入 Close 表中，并从 Open 表中删除。当 Open 表中第一个结点为终点时，找到最优路径，如

果 Open 表为空,则没找到最优路径。

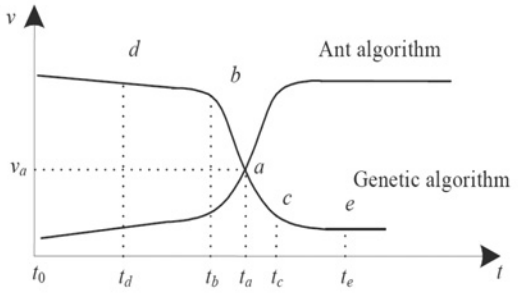

图 2 遗传算法与蚁群算法的速度 – 时间曲线

2.3 应用实例

利用 VC + +6.0 和 Mapx 5.0 作为实验环境,分别采用 Dijkstra 算法、原始 A* 算法和改进 A* 算法求路网中两结点的最短路,结果都能找到,但其搜索范围与时间却差别很大。如某市地图中,共有 3849 个结点,规定结点号沿经度升序方向增长,经度相同的沿纬度升序方向增长。以搜索结点 1747 到结点 1927 的最短路为例,图 3(a)为搜索结果,黑色道路为最短路;图 3(b – d)中黑色道路分别 Dijkstra 算法遍历道路、原始 A* 算法遍历道路和改进 A* 算法遍搜索程中遍历的道路,由图可见,Dijkstra 算法的搜索基本上是以起点为圆心以圆的方式向外拓展,A* 算法要优于 Dijkstra 算法,改进的 A* 算法最优。搜索结点 1 到结点 3840 之间的最短路,Dijkstra 算法的

搜索时间为 243ms,A* 算法的搜索时间为 89ms、改进的 A* 算法的搜索时间为 57ms。实验结果表明,改进 A* 算法要明显优于其它两种算法。

3 结论

对目前常用的最优路径规划算法进行了分析,比较了 Dijkstra 算法、A* 算法、B* 算法的效率,并将 A* 算法、遗传算法、蚁群算法有机融合,解决了基于 GIS 地最短路问题,通过实际测试,在效率和准确性方面都取得比较满意的效果。

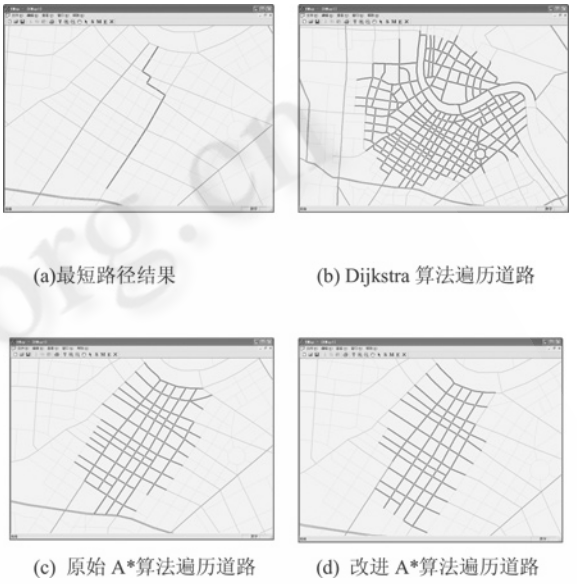

图 3

参考文献

1 章永进. 军用无人驾驶汽车基于电子地图导航、控制系统研究[硕士学位论文]. 天津: 军事交通学院, 2006,7.

2 徐业昌, 李树祥, 朱建民等. 基于地理信息系统的最短路径搜索算法[J]. 中国图像图形学报, 1998, 3 (1): 39 – 43.

3 刘光.. 地理信息系统二次开发教程 [M]. 北京: 北京航空航天大学出版社,2005.