

深圳应急指挥中心的最优路径规划技术研究^{*}

张春平¹, 关志超², 杨东援³

(1. 深圳中南运输集团公司, 广东 深圳 518049;

2. 中国电信—深大通信网络有限公司, 广东 深圳 518028;

3. 同济大学, 上海 200092)

摘要: 深圳应急指挥中心是一个集指挥、控制、通信、情报、刑侦、探测预警和综合保障为一体的现代化电子信息集成系统, 在使用最优路径规划技术上作了很好的尝试。从路径规划的基本原理入手, 介绍基于数据结构的最短路径搜索的高效实现方法, 并从缩小搜索空间和按地图分层分级搜索的应用策略出发分析路径规划算法。结合深圳的实际情况, 给出引入了基数堆、启发式搜索和分级搜索后的路径规划改进 Dijkstra 算法, 提供一个具有很高参考价值的规划技术应用研究。

关键词: 应急指挥中心; 最优路径; 规划; 基数堆; 改进 Dijkstra 算法

中图分类号: TP319 **文献标识码:** A **文章编号:** 0529-6579 (2003) S2-0160-05

深圳应急指挥中心的最优路径规划设计与实践, 在不久前抗击“非典”的战斗中, 发挥了巨大的作用。路径规划是帮助应急指挥中心的车辆驾驶员在出行前或行进中规划行驶路线的过程, 是车辆导航内的一个基本问题, 也是实现导航功能的前提条件。按照规划目的的不同, 路径规划又分为多车辆路径规划和单车辆路径规划, 多车辆路径规划适用于应急指挥中心的车队调度和交警交通管制, 单车辆路径规划适用于各种受命车辆导航系统。应急指挥车辆定位导航系统的路径规划属于单车辆规划的范畴, 要解决的主要问题是给定的道路网中寻找从出发点到目的地之间的最优路径。针对实际应用的不同要求, 在路径规划中可以采用最短行车距离、最少行车时间、最低通行费用等, 无论采用何种标准, 最优路径规划最终都归结为在待定道路网中寻找具有最小代价的最短路径问题, 即图论中的最短路径问题。经典 Dijkstra 最短路径算法是求解这一问题的基本方法, 而且已有针对 Dijkstra 算法的具体实现。但是, 在深圳应急指挥中心车辆导航中不能直接使用, 因为在实际应用中记录路网信息的电子地图数据库规模很庞大, 而负责路径规划功能的导航计算机受车载环境和成本限制, 处理能力和系统存储资源都十分有限, 难以承担苛刻的计算量需求。我们在实际应用中, 综合利用深圳市区电子

地图的数据结构、启发式搜索算法和多级搜索技术, 从而有效地加速最优路径的搜索。

1 最短路径问题

图论是研究与图相关的理论和算法的科学, 近年来随着计算机技术的飞速发展, 其在计算科学、系统科学、数据结构、网络设计等方面的作用正变得越来越显著, 应急指挥中心的车辆导航系统的最优路径规划就属于图论中的最短路径问题。

给定带权有向图 $G = (V, \{E\})$, 其中 V 是包含 n 个顶点的顶点集, E 是包含 m 条弧的弧集, $\langle u, w \rangle$ 是 E 中从 u 至 w 的弧, $c(u, w)$ 是弧 $\langle u, w \rangle$ 的非负权值, 设 u_s, u_t 为 V 中的顶点, $P_{st} = \{u_0 = u_s, u_1, \dots, u_n = u_t\}$ 为 V 中由 u_s 到 u_t 的一条路径, 则路径 P_{st} 的权值总和可表示为:

$$TW_{P_{st}} = \sum_{i=0}^{n-1} C(u_i, u_{i+1})$$

所谓最短路径问题就是指在带权有向图中, 寻找从指定起点到终点的一条具有最小权值总和的路径问题, 如果把权值看成是弧的长度属性(距离), 那么目标路径就是从起点到终点的最短路径; 如果把应急指挥中心路径规划中的优化标准量化为道路的出行代价, 则最优路径规划就可以归结为在特定道路网中寻找具有最小出行代价总和的最优路径问

* 收稿日期: 2003-08-12

基金项目: 深圳市政府抗击“非典”重点实施工程项目 (2003)

作者简介: 张春平 (1961 年生), 男, 总经理, 同济大学博士研究生; 关志超 (1959 年生), 男, 技术总监, 同济大学博士研究生; 杨东援 (1955 年生), 男, 常务副校长, 教授, 博士生导师; Email: guanzhichao@

mail.china.com

题, 显然这与最短路问题是等价的。在这里我们为了说明最短路求解算法, 首先来讨论单源点最短路问题, 即给定带权有向图 $G = (V, \{E\})$ 和源点 u , 求 u 至 G 中其余各顶点的最短路径。关于这一问题, Dijkstra 提出了按路径长度递增的次序产生最短路径的算法 (即经典 Dijkstra 算法), 其产生最短路径的原理如下:

设置一个辅助向量 D , 它的每个分量 d_i 表示当前所找到的从起始点 u 到每个终点 u_i 的最短路径的长度, 设置 D 的初始状态为: 若从 u 到 u_i 有弧, 则 d_i 为弧上的权值; 否则令 d_i 为 ∞ 。显然, 长度为:

$$d_i = \min_i \{d_i \mid u_i \in V\}$$

的路径就是从 u 出发的长度最短的一条路径。设 S 为已求得最短路径的终点的集合, 则可以证明: 下一条最短路径 (终点为 x) 或者是弧 $\langle u, x \rangle$ 、或者是中间只经过 S 中的顶点而最后到达顶点 x 的路径; 因此, 下一条长度次短的最短路径的长度必为:

$$d_i = \min_i \{d_i \mid u_i \in S\}$$

其中, d_i 或者是弧 $\langle u, u_i \rangle$ 上的权值, 或者是 $d_k (u_k \in S)$ 和弧 $\langle u_k, u_i \rangle$ 上的权值之和。根据以上原理可得到如下描述的最短路径改进算法:

(1) 利用邻接矩阵 C 来表示带权有向图 G , 其元素 c_{ij} 表示弧 $\langle u_i, u_j \rangle$ 上的权值; 若弧 $\langle u_i, u_j \rangle$ 不存在, 则将 c_{ij} 设为 ∞ 。令 S 为已找到从 u_s 出发的最短路径的终点的集合, 将其初始化为空集。用 d_i 表示从 u_s 出发到终点 u_i 的可能达到的最短距离, 取其初始值为: $d_i = c_{si}, u_i \in V$ 。

(2) 选择 u_j , 使得: $d_j = \min_i \{d_i \mid u_i \in S\}$, 则 u_j 就是当前从 u_s 出发的最短路径的终点。令: $S = S \cup \{U_j\}$ 。

(3) 修改从 u_s 出发到集合 $V \setminus S$ 中任一顶点 u_k 可达最短路径长度, 如果 $d_j + c_{jk} < d_k$, 则修改 d_k 为: $d_k = d_j + c_{jk}$ 。

(4) 重复操作 (2)、(3) 共 $n-1$ 次, 由此求得按路径长度递增次序排列的从 u_s 出发至图 G 中其余各顶点的最短路径序列。

以上算法将产生从源点出发至其余各顶点的最短路径, 对于应急指挥中心路径规划而言, 我们只需要从源点至指定终点的一条路径, 为此可将算法作简单的修改: 即设目标路径终点为 u_t , 则在操作 (2) 中, 如果发现 $u_j = u_t$, 算法即终止; 此时与 d_t 对应的路径就是从源点 u_s 至终点 u_t 的最短路径。

2 基数堆结构 Dijkstra 算法的高效实现

堆结构是由若干数据项组成的数组, 其中的每个数据项都拥有一个属性值。若弧权值 $c(u, w)$ 只取整数值且具有上界 C , 则采用基数堆 (radix haeps) 结构存放已标记顶点能够缩减 Dijkstra 算法的时间复杂度, 基数堆改进 Dijkstra 算法执行效率的依据是顶点权值 $d(u)$ 的下列性质:

(1) 对任意的顶点 u , 若 $d(u) < \infty$, 则有: $d(u) \in [0, \dots, nC]$;

(2) 对任意已标记顶点 $u \neq u_s$, 有: $d(u) \in [d_{(x)}, \dots, d_{(x)} + C]$; 其中 x 是在最近一次扫描操作中产生已扫描顶点。

基数堆是 $B = \log_2(C+1) + 2$ 个带序号桶结构的集合, 每个桶存放权值在特定范围内的已标记顶点。对于序号为 i 的桶, 定义其关联范围如下: $\text{size}(1) = 1, \text{size}(i) = 2^{i-2}, 2 \leq i \leq B-1, \text{size}(B) = nC+1$ 为每个桶指定一个权值上界 u , 则序号为 i 的桶内所存放的顶点权值范围表示为: $\text{range}(i) = [u(i-1), \dots, \max\{u(i-1) + \text{size}(i), u(i)\}]$, 取 u 的初始值为: $u(0) = 0, u(i) = 2^{i-1-1}, 1 \leq i \leq B-1, u(B) = nC+1$, 则堆结构中的顶点权值范围为 $[0, \dots, nC+1]$ 。在 Dijkstra 算法执行过程中, 每个桶结构的关联范围保持不变, 上界 u 则随扫描过程的进行作动态调整。此时堆结构的三种操作执行过程如下:

① $\text{insert}(h, x)$: 从 $i = B-1$ 开始按递减次序检测 $u(i)$, 若 $u(i) < d(x)$, 则将 x 存入桶 $i+1$;

② $\text{delete}(\min(h))$: 从 $i = 1$ 开始按递增顺序检测桶 i , 并将第一个非空桶内的最小权值顶点作为操作结果返回;

③ $\text{decrease}(h, x, \text{value})$: 令 $d(x) = \text{value}$, 设 x 在桶 j 内, 若 $d(x) \in \text{range}(j)$, 操作结束; 否则从桶 j 中删除 x 并由 $i = j-1$ 开始重新执行 $\text{insert}(h, x)$ 。

在执行 delete 操作后, 还要修改桶结构的权值上界并重新调整已标记顶点在整个堆结构内的分布, 这也是整个算法中最复杂的操作。设 $u = \text{delete}(\min(h))$, 首先对 u 作如下修改: $u(0) = d(u) - 1, u(1) = d(u), u(i) = \min\{u(i-1) + \text{size}(i), u(j), 2 \leq i \leq j-1\}$, 式中 j 是 u 所在桶的序号, 然后将桶 j 中的所有已标记顶点全部移出并重新执行 insert 操作。考虑到桶的关联范围有下列关系:

$$\sum_{i=1}^{j-1} \text{size}(i) \geq \min\{\text{size}(j), c+1\}, 2 \leq j \leq B$$

连同性质 (2) 一起保证了 j 中的每一个顶点都能被移入序号更小的桶内。修改后桶结构内的顶点权值范围变为 $[d(u), \dots, nC + 1]$ 。采用基数堆结构后, delete 操作的内容变为两项:

① 在 $[d(u), \dots, d(u) + C]$ 范围内寻找具有最小权值的数据项, 执行时间复杂度为 $O(\log_2 C)$;

② 整理已标记顶点在桶结构中的分布, 相当于为每个待调整顶点增加一次 insert 操作。

对任意一个已标记顶点来说, 其所处的桶结构序号只可能按严格递减的次序变动, 即每顶点最多变动 $B - 1$ 次; 因此将正常的 insert 操作、由 delete 附加的 insert 操作和由 decrease 附加的 insert 操作加在一起, 其总的执行时间复杂度也不会超过每顶点 $O(B - 1)$ 。由此可以计算出采用基数堆结构实现 Dijkstra 算法所耗费的总时间复杂度为(忽略常数项和比例系数):

$$O(n(B - 1)) + O(n \log_2 C) + O(m - n + 1) = O(m + n \log_2 C)$$

即在采用基数堆结构后, Dijkstra 算法的时间复杂度由线性时间减少到对数时间。

3 快速最优路径规划算法

对于应急指挥中心, 车辆导航系统中的最优路径规划问题属于图论中的最短路问题, 但是它同样具有自己的特点。首先, 车辆导航的实时性要求是显而易见的, 在出行过程中一旦由于路况变化或其他原因使车辆未能按预定导航路线行驶, 则系统必须重新计算最优路线, 因此对于规划算法的执行效率要求较高, 即运行速度一定要快。另外在实际应用中, 导航深圳市区电子地图的规模十分庞大, 使得执行路径规划算法所需要的时间和存储空间要求都变得非常苛刻; 而负责处理路径规划任务的导航计算机由于受成本和车载环境的限制, 其处理能力和系统存储资源均十分有限, 难以负担繁重的计算任务。

为了适应应急指挥的车辆导航要求, 路径规划算法的高效实现我们利用三种方法来获得: ①采用先进的数据结构缩短运行时间; ②采用先进的搜索技术减小搜索空间; ③采用地图分层和分级搜索技术控制规划路网的规模。

3.1 启发式路径规划方法

在采用矢量编码方式的深圳市区电子地图数据库中, 道路以直线或折线方式存储, 并表示为一系坐标点的有序集。为了获得适用于路径搜索的路网图, 必须将包含交叉点的道路拆分成最基本的路段, 使其只在端点处与其他路段相交; 拆分后的基

本路段对应于路网图中的弧, 其端点就是图中的顶点。应急指挥车辆可双向行驶的路段要用两条端点相同、方向相反的弧表示; 路网图的拓扑关系可以用两个数组表示, 其中分别存放与顶点相关的信息和与弧相关的信息, 并按顶点的编号次序排序以方便查找; 与路网图拓扑关系无关的其它道路属性信息用单独的数组表示, 数组间存在链接关系。Dijkstra 算法是按长度递增的次序产生最短路径, 凡是到源点的最短路径长度小于目标路径长度的顶点都会被搜索到, 因此算法的搜索空间可以近似看成是圆形。如果在搜索中能够优先考虑靠近终点方向的顶点, 那么对于减小算法的搜索空间无疑是有帮助的。

启发式搜索是指在路径规划中引入启发信息以提高搜索效率的方法, 其算法是目前广为流行的路径规划启发式算法, 对实现道路网的最佳优先搜索十分有效。通过选择合适的估价函数, 其算法寻找最优路径所使用的搜索空间比经典 Dijkstra 算法小。采用启发信息的目的是提供一个顶点距离目标点有多远的估计, 以使搜索算法优先考虑最有希望的顶点。定义顶点 u 的启发式估价函数 $f'(u)$ 为:

$$f'(u) = g(u) + h'(u)$$

式中, $g(u)$ 是从起点 S 到当前顶点 u 的实际费用, $h'(u)$ 是从当前顶点 u 到目标顶点 t 的费用估计函数, 则其算法优先搜索具有最小 $f'(u)$ 值的顶点。 $h'(u)$ 的选择取决于路径规划的标准, 在搜索最短路径时, 以顶点权值 $d(u)$ 作为起点到当前顶点实际费用, 以当前顶点到目标顶点的直线距离 $d'(u)$ 作为费用估价函数, 则有:

$$f'(u) = d(u) + d'(u)$$

将 Dijkstra 算法中的顶点权值 $d(u)$ 用 $f'(u)$ 代替, 即实现了启发式最短路径搜索。

3.2 基于多层地图的分级搜索

从 Dijkstra 算法的时间复杂度分析可以看出, 深圳路网图的规模是决定最优路径搜索时间的关键因素。如果用合理的方法减少搜索所涉及的顶点和弧的数量, 将会使路径规划获得有效地加速。采用多层地图和分级搜索技术可实现对路径搜索空间的控制, 我们的做法是将深圳市区道路网用高低两种分辨率的电子地图表示, 首先在低分辨率图层中进行全局搜索以获得大致的目标路径, 然后利用该路径将高分辨率图层分解为一系列局部搜索空间进行搜索, 并将获得的局部最优路径合并在一起得到最终的目标路径。假设有包含深圳等级公路信息的路网图 G , 则分级搜索的过程可以描述如下:

(1) 将 G 中二级以上高等级公路抽出单独用一

个图层 V 表示,并将起点 s 与终点 t 加入到图层 V 中。

(2) 在 V 中搜索从 s 到 t 的最优路径 $R_{st}=(u_1=s, e_1, u_2, e_2, \cdots, u_q, e_q, u_{q+1}=t)$,其中 u_i 是路径所经过的顶点, e_i 是连接相邻顶点的弧。

(3) 将 R_{st} 划分为 p 段,令其中第 i 段的端点为 u_{s_i} 和 u_{t_i} ,且 $u_{s_1}=s, u_{t_p}=t, u_{s_i}=u_{t_{i-1}}, i>1; u_{t_i}$ 按如下条件选取:

$$\sum_{j=s_i}^{t_{i-1}} e_j \leq L, i \leq p, \sum_{j=s_i}^{t_i} e_j > L, i < p$$

式中, L 为预定的长度划分界限。

(4) 从 $i=1$ 开始在 G 中搜索由 u_{s_i} 至 u_{t_i} 的最优路径 R_i' ,并将获得的局部最短路径合并生成全局目标路径 $R_{st}'=(R_1', \cdots, R_p')$ 。

在大多数情况下,还可以采用顶点过滤来进一步加快 G 图层的搜索速度,设 G 中顶点 u 的坐标为 $u(x,y)$,则在搜索 R_i 时仅考虑坐标在如下范围内的顶点:

$$\begin{aligned} x_{\min} &\leq u(x) \leq x_{\max}, \\ y_{\min} &\leq u(y) \leq y_{\max} \end{aligned}$$

式中的过滤参数定义如下:

$$\begin{aligned} x_{\min} &= \min(u_{s_i(x)}, \cdots, u_{t_i(x)}) - \mu L \\ x_{\max} &= \max(u_{s_i(x)}, \cdots, u_{t_i(x)}) + \mu L \\ y_{\min} &= \min(u_{s_i(y)}, \cdots, u_{t_i(y)}) - \mu L \\ y_{\max} &= \max(u_{s_i(y)}, \cdots, u_{t_i(y)}) + \mu L \end{aligned}$$

其中参数 μ 是保证在实行顶点过滤后仍有最优路径存在的扩展因子。

采用以上分级搜索方法后,原来的搜索空间将被一系列小的局部搜索空间代替。当路网图规模比较大时,这样处理将会使搜索空间得到显著的缩减。另外一旦道路交通状况发生了变化,例如某条公路被临时关闭,常规的规划算法必须要在权值更新后的路网图中重新搜索从源点到终点的最优路径;而在采用分级搜索后,我们只需要更新包含变化路段的一段局部最优路径,由于这一过程只会涉及一个小的局部搜索空间,因此重新规划最优路径的速度会大大加快。在深圳应急指挥中心,低分辨率图层 V 的构造可以在路径规划前预先完成。需要指出的是,采用分级搜索后所获得的最优路径实际上是局部最优路径解的叠加,它不一定是真正的

全局最优路径解,但是足以满足路径导航的需要。

4 结 语

在深圳应急指挥中心,为了检验快速路径规划算法的有效性,我们采用 Visual C⁺⁺ 编制了最优路径搜索程序并进行仿真试验。程序分别按以下四种方法搜索最短路径:①基于堆结构的经典 Dijkstra 算法;②应用基数堆结构的改进 Dijkstra 算法;③应用基数堆结构的启发式算法;④综合应用了分级搜索与基数堆结构的启发式算法。

搜索地图采用深圳罗湖区老街街道图,其中共包含 8 952 条弧和 3 478 个顶点;为应用基数堆结构,我们将弧长度以 50 m 为基本单位分成 512 个级别。在分级搜索中,划分长度 L 取为 5 000 m,同时起用定点过滤,扩展因子 μ 取为 0.1,如果局部搜索结束后没有获得最短路径解,则关闭顶点过滤重新搜索。仿真程序在主频 75 MHz、配备 20 M 内存的车载计算机上进行了 10 次穿越全图的长距离最短路径搜索,结果表明搜索顶点总数是搜索结束后的已标记顶点和已扫描顶点总数之和,低分辨率图层是建立在搜索程序运行前预先完成,其处理时间不统计在运行时间内。由此可见,在引入了基数堆、启发式搜索和分级搜索后,路径规划算法得到了有效加速,大大地提高了应急指挥中心实战应用中的快速反应能力。

参考文献:

[1] HUANG Y H, RUNDENSTEINER E A, JING N. Evaluation of hierarchical path finding technologies for ITS route guidance[J]. Annual Meeting of ITS American, 1996: 340—350.

[2] HUSNI H. Cellular Mobile Radio Systems, Designing Systems for Capacity Optimization[M]. British: John Wiley & Sons, 1997.

[3] STAIR R M, REYNOLDS G W. 信息系统原理[M]. 北京:机械工业出版社,2000.

[4] 杨东援. 交通规划决策支持系统[M]. 上海:同济大学出版社,1997. 5.

[5] 郑人杰. 计算机软件测试技术[M]. 北京:清华大学出版社,1992.

[6] 竺南直,朱德成. 指挥自动化系统工程[M]. 北京:电子工业出版社,2002.

Optimum Route Planning Technology Research of
Shenzhen Emergency Command Center

ZHANG Chun ping¹, GUAN Zhi chao², YANG Dong yuan³

- (1. Shenzhen Zona Transportation Group Co., LTD., Shenzhen 518049, China;
- 2. China Telecom Shenda Communication Network Co., LTD. Shenzhen 518028, China;
- 3. Tongji University, Shanghai 200092, China)

Abstract: As a modern electronic information integration system, Shenzhen emergency command center is capable of many functions, such as command, control, communications, information, scout, forewarning, indentification, etc, and has made a good try of the application of optimum route planning technology. According to the principle of route planning, we analyze the route planning algorithm in terms of compressing searching space and hierarchical layered map, and give an efficient resolvent of shortcut searching based on data structure. Moreover, integrating the fact of Shenzhen, this article introduces a route planning ameliorate Dijkstra algorithm interlards radix heaps, heuristic searching and hierarchical searching. Thus this is a valuable reference to planning technology application research.

Key words: emergency command center; optimum route; planning; radix heaps; ameliorate Dijkstra arithmetic