

用 $\Theta(t)$ 的广义连接图求有障碍时的最短路径*

周智^{1,2+}, 蒋承东^{1,2}, 黄刘生^{1,2}, 顾钧³

¹(中国科学技术大学 计算机科学与技术系, 安徽 合肥 230027)

²(国家高性能计算中心(合肥), 安徽 合肥 230027)

³(香港科学技术大学 计算机科学系, 香港)

Finding Obstacle-Avoiding Shortest Path Using Generalized Connection Graph with $\Theta(t)$ Edges

ZHOU Zhi^{1,2+}, JIANG Cheng-Dong^{1,2}, HUANG Liu-Sheng^{1,2}, GU Jun³

¹(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

²(National High Performance Computing Center at Hefei, Hefei 230027, China)

³(Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, China)

+ Corresponding author: Phn: 86-551-3603747, Fax: 86-551-3601013, E-mail: cdjiang@mail.ustc.edu.cn

<http://www.nhpc.ustc.edu.cn>

Received 2002-04-02; Accepted 2002-06-10

Zhou Z, Jiang CD, Huang LS, Gu J. Finding obstacle-avoiding shortest path using generalized connection graph with $\Theta(t)$ edges. *Journal of Software*, 2003,14(2):166~174.

Abstract: Finding obstacle-avoiding shortest path is an important problem in VLSI design, and connection graph is a fundamental tool to resolve the problem. The known best graph grounded on knowledge of free area, and it has $O(t)$ vertices and $O(t \log t)$ edges, in which t is the number of extreme edges of the obstacles. In this paper, a generalized connection graph G_G is introduced, which is derived from some new conception such as generalized free area. G_G is made up with vertex that represents the generalized free area and edges for their adjacency. It has only $\Theta(t)$ vertices and $\Theta(t)$ edges, and it is planar graph. An $O(t \log t)$ time algorithm using plane scanning is designed to construct G_G , and the do not change direction heuristic together with A^* algorithm is used for getting the shortest obstacle-avoiding path using G_G through the conception of formal path. This algorithm shorten the time complexity from $O(t \log t)$ to $\Theta(t)$.

Key words: VLSI design; generalized free area; generalized connection graph; shortest path; NP-hard

摘要: 在有障碍时求两点间的最短路径是 VLSI 设计、机器人设计等领域中的基本问题,连接图是研究此问题的基本工具.现有算法构造的最好的连接图 G_F 是基于自由区的概念而设计的,其顶点数和边数分别为 $O(t)$ 和 $O(t \log t)$,其中 t 为障碍的极边数.提出了广义自由区和极大正规划分的概念,在此基础上得到广义连接图 G_G ,用来表征广义自由区之间的邻接情况,其顶点数和边数均为 $\Theta(t)$,且具有平面图性质.同时还提出了基于扫描线的极

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1998030401 (国家重点基础研究发展规划(973))

第一作者简介: 周智(1976—),男,湖南娄底人,博士生,讲师,主要研究领域为复杂性理论,算法设计.

大正规划分构造算法,其时间复杂度为 $O(t \log t)$;并提出规范路径的概念,以及采用“不改向”启发式策略的 A^* 算法在广义连接图 G_G 中寻找两点间的最短路径,算法的时间复杂度由基于 G_F 的现有算法的 $O(t \log t)$ 降低到 $\mathcal{O}(t)$.

关键词: VLSI 设计;广义自由区;广义连接图;最短路径;NP-hard

中图法分类号: TP301 文献标识码: A

在有障碍时求两点间的最短路径是机器人设计、VLSI 布线设计和地理信息系统中的基本问题.基于设计规则和工艺的要求,VLSI 设计中的布线问题有以下特性:

- (1) 布线空间为 $m \times n$ 的有限平面网格;
- (2) 电路元件和已布好的线成为新的布线障碍;
- (3) 两点间路径和障碍的边均为水平或竖直边.

现有的算法分为两类:走迷宫(maze-running)算法和线搜索(line-search)算法.

Lee 算法^[1]为第 1 个走迷宫算法,采用的是广度优先搜索方法,此算法的缺点在于其时间和空间复杂度均为 $O(mn)$,且每个节点需 $O(\log L)$ 字节存储空间,其中 L 为源节点到目标节点的最短路径长度.Lee 算法有许多改进版本:Akers^[2]引进一种编码策略使每节点的存储空间只需 2 字节而与 L 无关;Soukup^[3]提出一种将深度优先搜索与广度优先搜索相结合的方法,以“不改变方向”为启发策略提高算法性能,但无法保证得到最优解.

迷宫算法基于单位格点搜索,故在时间和空间上是低效的.为此,人们提出了线搜索算法.基本思想是构造一个代表障碍和节点位置的图,通过在图中寻找最短路径得到原格点中的最短路径.此类算法有以下特点:

- (1) 构造的图比原单元格点图稀疏,因此而致算法时间复杂度的降低;

(2) 构造的图中路径与原网格中的路径可以对应起来.对源和目标节点,构造的图中存在与原网格中的最短路径相对应的路径.

此类算法较早由 Hightower^[4]所提出.后来有一些较新的线搜索算法是基于计算几何的方法,如 Clarkson^[5]等,计算几何算法的时间复杂度较低但实现过于复杂,因而难以在实际中应用.Zheng^[6]引入了一个强连接图,其算法时间复杂度为 $O(e^2)$,其中 e 为所有障碍的总边数,它同时采用 A^* 算法和“不改向”启发策略,有较好的实用性.Wu^[7]引入一个很小的连接图,称为轨迹图(track graph),它不是强连接图,但对例外情况有简单而有效的处理.该算法的时间和空间复杂度分别为 $O((e+k) \log(t))$ 和 $O(e+k)$,其中 t 为障碍的极边数, k 为轨迹的交点数,且 $k=O(t^2)$.在最坏情况下, $t=e, k=e^2$.文献[8,9]中给出了基于自由区定义的连接图,其顶点数和边数分别为 $O(t)$ 和 $O(t \log t)$,为现有的线搜索算法中复杂度最低的一种.

本文第 1 节给出了基本的定义、特性,并简要回顾了几种关键性的连接图.第 2 节定义了广义自由区、广义划分、正规划分、极大划分等概念,在此基础上定义了广义连接图 G_G ,并证明了其正确性和复杂度.第 3 节给出基于扫描线的构造算法.第 4 节给出了两点间的规范路径的定义及求最短路径的 A^* 算法.第 5 节对实验结果进行了比较和分析.最后是总结.

1 基本定义及特性

在以下的讨论中,取平面 R^2 为格点的定义域空间,所得到的结果对于 $m \times n$ 网格是适用的.平面上的距离为 Manhattan 距离,即点对 $p_1(x_1, y_1), p_2(x_2, y_2)$ 之间的距离为 $\rho(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$.

定义 1(路径). 连续函数 $P: t \in [0, 1] \rightarrow (x, y) \in R^2$ 为平面上的路径, $P[a, b]$ 表示与 $[a, b](0 \leq a < b \leq 1)$ 对应的子路径.若 $\forall a < t_1, t_2 < b, y(t_1) = y(t_2)$,则称 $P[a, b]$ 为水平的;若 $\forall a < t_1, t_2 < b, x(t_1) = x(t_2)$,则称 $P[a, b]$ 为竖直的.若 $\exists n, 0 = t_1 < t_2 < \dots < t_n = 1, s.t. P[t_i, t_{i+1}](1 \leq i < n)$ 为水平的或竖直的,则称 P 为正(rectilinear)路径,并称 $T = (t_1, \dots, t_n)$ 为 P 的一个框架. P 的最小框架是指使 n 最小的框架.以下除非特别说明,路径均指正路径,框架均指最小框架.若 $T = (t_1, \dots, t_n)$ 为路径 P 的框架,则称 $P(t_i)$ 为 P 的顶,子路径 $E_i = P[t_i, t_{i+1}]$ 为 P 的边.定义边的方向为

$$Dirt(E_i = P[t_i, t_{i+1}]) = \begin{cases} R, & \text{若 } E_i \text{ 水平, 且 } x(t_i) < x(t_{i+1}) \\ L, & \text{若 } E_i \text{ 水平, 且 } x(t_i) > x(t_{i+1}) \\ U, & \text{若 } E_i \text{ 竖直, 且 } y(t_i) < y(t_{i+1}) \\ D, & \text{若 } E_i \text{ 竖直, 且 } y(t_i) > y(t_{i+1}) \end{cases}$$

若 P 上只有一条边, 则称此路径为线段. 令 $L(P) = \sum_{i=1}^{n-1} \rho(P(t_i), P(t_{i+1}))$ 为路径 P 的长度. 若 $P(t) = P(t')$ 必有 $t = t'$ 或 $\{t, t'\} = \{0, 1\}$, 则称 P 为简单路径. 对简单路径 P , 若 E_{i-1} 和 E_{i+1} 方向不同, 则称 E_i 为路径 P 上的极边. 简单路径有如下性质:

性质 1. 对任意正路径 P , 最小框架是惟一的, 且 E_i 与 E_{i+1} 不同向.

性质 2. 令 P 为两点间的路径, 若其所有水平边与竖直边分别保持方向不变, 则 P 为一条最短路径.

性质 3. 两点间无极边的路径必为最短路径, 且其长度为两点间的距离.

定义 2(多边形). 对于路径 P , 若 $P(0) = P(1)$, 即其起点与终点重合, 则称 P 为闭合路径, 此时定义 $t_{n+1} = t_2, \dots$; 一条简单闭合路径将平面划分为两部分: 有界的部分称为内部, 记为 $in(P)$; 无界的部分称为外部; 定义一个多边形 F 为简单闭合路径 P 和其内部的并, 边为 P 的边, 它的顶为 P 的顶, P 为此多边形的边界, 记为 $bound(F)$, 令 $in(F) = in(P)$; 定义边界上的极边为多边形的极边 (下文中将不再说明极边是多边形的极边还是普通的路径上的极边), 记其集合为 $ex(F)$, 极边上的顶称为极顶, 记其集合为 $exv(F)$; 称包含多边形的最小矩形区为此多边形的极区 $ExR(F)$. 显然, 对于凸多边形 F , $ExR(F) - F$ 最多由 4 个不连通的多边形构成, 称之为空极区.

定义 3(凹凸形). 对于平面上的点集 $S, \forall p_1, p_2 \in S$, 若 $y(p_1) = y(p_2)$, 则对 $\forall p \in \overline{p_1 p_2}$, 有 $p \in S$, 则称 S 为 X 凸的. 同样定义 Y 凸. 若 S 既为 X 凸的又为 Y 凸的, 则称 S 为凸的. 否则称为凹的. 对多边形 F , 若 F 为凸的, 则称 F 为凸多边形, 否则为凹多边形.

性质 4. Manhattan 空间中的任意凸多边形 F 有且仅有 4 条极边, 且这 4 条极边等价于 $ExR(F)$ 的边界与 F 的边界之交, 其方向分别为 L, R, U, D , 分别称之为 F 的左、右、上、下极边.

性质 5. 对于凸多边形 F 为障碍的情况^[8,9], F 同一空极区中的两点间存在不含极边的路径, 即最短路径; 否则, 两点 s, t 分别属于不同空极区, 则对连接 s, t 的最短路径 P 上的极边 e , 必存在 F 上的极边 e_f , 使得 $e_f \subseteq e$, 且 $P \subset ExR(F)$.

定义 4(问题定义). 在实际问题中, 问题空间中的点不是连续的, 而是有限间隔的网格. 该网格对应于图 $G_0 = (V_0, E_0)$, 其中 $V_0 = \{(i\delta_x, j\delta_y) | i, j \in Z\}$, 且 $e = \langle u, v \rangle \in E_0 \Leftrightarrow \rho(u, v) = \delta_x$ 或 δ_y . 不妨设 $\delta_x = \delta_y = 1$, 则该网格对应于平面上整数点的网格 N^2 , 故不失一般性, 以下用 N^2 取代 R^2 , 同时可将上面定义的关于路径、边、多边形、凹凸的概念对应到网格上. 定义 $SPP = (OB, PIN)$ 上, 其中 $OB = \{ob_i | ob_i \subset V_0, \text{且 } ob_i \text{ 在 } G \text{ 中的生成子图连通}\}$ 为障碍集合, 记 $V_0 - \bigcup_{ob_i \in OB} ob_i$ 为 $N^2 - OB$, 则 $PIN \subset N^2 - OB$ 为目标节点集合, $G_{SPP} = (V_{SPP}, E_{SPP})$ 为 G_0 在 $N^2 - OB$ 上的生成子图, 则在 G_{SPP} 上求 PIN 中的点对之间的最短路径的问题为 Manhattan 空间中有障碍时点对之间的最短路径问题. 不失一般性, 我们限制障碍为凸多边形 (凹多边形的情况见文献[6]). 同时, 为描述方便, 设 G_{SPP} 定义在 $N^2 - OB'$ 上, 其中 $OB' = \bigcup_{ob_i \in OB} in(ob_i)$.

定义 5(连接图). 对图 $H = (V, E)$ 和图 $G = (V, E)$, 若 V' 为 V 的子集, 存在 G 的一个连接 V' 的子图与 H 是同构的, 且 H 是连通的, 则称 H 为 G 中 V' 的连接图, 简称 H 为 G 的连接图. 若 H 为 G 的连接图, 且任意 V' 中两点的最短路径长度在 H 和 G 中是相同的, 则称 H 为 G 的强连接图.

线搜索算法的基本思想是在强连接图中寻找最短路径, 因而连接图是算法性能的关键. 为与典型的线搜索算法进行比较, 以下我们先简单介绍 3 种最典型的线搜索算法使用的连接图.

定义 6(基于轨迹的连接图). 若两点集有两个以上的交点, 则称此两点集为交叠的. 对 $e \subset N^2 - OB$, 若有线段 l 与 e 交叠, 且不存在与 e 交叠的线段 l' 使得 $l \subset l'$, 则称 l 为 e 在 $N^2 - OB$ 上的极大轨迹, 简称轨迹, 记 $T(e)$ 为边 e 的轨迹集合, $T(ob_i)$ 为障碍 ob_i 的所有边的轨迹集合. 令所有的轨迹集合为 $L_C = (\bigcup_{ob_i \in OB} T(ob_i)) \cup (\bigcup_{p \in PIN} T(\{p\}))$, 其中 $T(\{p\})$ 为单个点 p 的轨迹集合, 也就是经过 p 的水平竖直的两条轨

迹.定义边连接图^[6] $G_C = (V_C, E_C)$, 其顶集 V_C 为 L_C 中线在 $N^2 - OB$ 中的交点的集合, 边集 E_C 为 L_C 中线段被交点所分割成的子线段的集合, 如图 1(a) 所示. $L_{ex} = (\bigcup_{ob_i \in OB} T(ex(ob_i)))$ 为极边的轨迹集合, 令 $T'(p)$ 为包含点 p 且端点在 L_{ex} 或 $\bigcup_{ob_i \in OB} bound(ob_i)$ 上的极短线段, 令 $L_T = L_{ex} \cup (\bigcup_{p \in PIN} T'(p))$, 定义轨迹连接图^[7] $G_T = (V_T, E_T)$, 其中 V_T 为 L_T 中线的交点的集合, 边集 E_T 为 L_T 中线段被交点所分割成的子线段的集合, 如图 1(b) 所示.

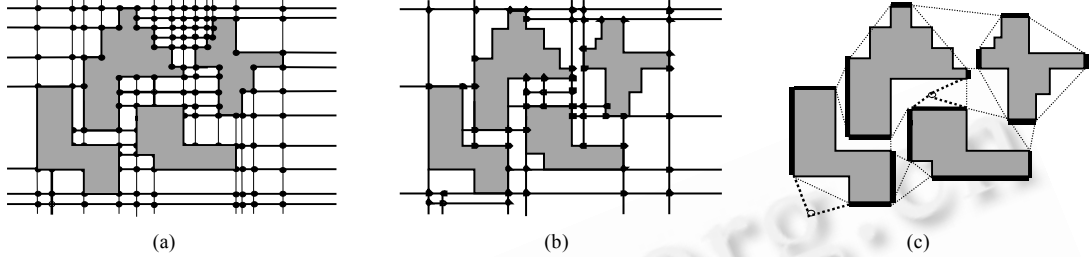


Fig. 1 Three kinds of typical connection graphs

图 1 3 种典型的连接图

定理 1^[6]. G_C 为 G_{SPP} 的强连接图.

定理 2^[7]. 若 P 为 s 到 t 的最短路径且至少有一条极边, 或无极边但 P 至少与 OB 中的一条水平极边轨迹和一条竖直极边轨迹相交, 则在 $G_T(OB, \{s, t\})$ 中有由 s 到 t 的同样长的路径.

定义 7(基于自由区的连接图). 对于一个不是完全包含在障碍中的矩形区, 若其边界的两对角顶为不同障碍的极顶, 且在边界上无其他极顶, 则称之为自由极边区, 简称自由区(记其集合为 $Free(G_{SPP})$), 其边界上的极顶为此自由区的特征顶. 将点 p 视为退化的障碍多边形 $ob(p)$, 令 $OB_0 = OB \cup \{ob(p) \mid p \in PIN\}$, 则定义自由区连接图^[9] $G_F = (V_F, E_F)$, 其中 $V_F = \bigcup_{ob_i \in OB_0} exv(ob_i)$, E_F 中边连接一条极边的两个极顶点或者连接一个自由区的特征顶点, 如图 1(c) 所示.

性质 6. 自由区中的任意两点的最短路径在自由区内, 且其长度等于两点间的距离.

定理 3^[9]. G_F 为 G_{SPP} 的强连接图.

2 广义自由区和 G_G

显然, 性质 6 是自由区连接图的基础, 用区域取代路径使得 G_F 减小了连接图上的顶点和边的数量, 且提高了求解效率. 注意到传统的连接图上顶点必须为极点或网格点, G_F 也遵守了这一原则, 使其构造受到限制, 造成如下的缺陷: (1) G_C 和 G_T 是平面图, 而 G_F 不是; (2) 自由区的定义局限于矩形, 限制了更简洁的构造方式. 为此, 我们拓展自由区的定义, 试图构造新的表示方法.

定义 8(广义自由区). 对 $N^2 - OB$ 上的点集 F_g , 若其中任意两点之间的最短路径的长度均等于其距离, 则称其为 $N^2 - OB$ 上的广义自由区, 记 $N^2 - OB$ 上所有的广义自由区的集合为 $FG(N^2 - OB)$.

性质 7. $F_g \in FG(N^2 - OB) \Leftrightarrow F_g$ 为 $N^2 - OB$ 上的凸多边形.

这个性质保证广义自由区在算法上是可以得到的. 进一步有如下定义:

定义 9(广义自由区划分). 对 $P_g \subseteq FG(N^2 - OB)$, 若 $\bigcup_{F_g \in P_g} F_g = N^2 - OB$, 且 $\forall F_1, F_2 \in P_g, (F_1 \cap F_2) \cap (\text{in}(F_1) \cup \text{in}(F_2)) = \emptyset$, 即其中任意两个区域只在边界部分相交, 则称 P_g 为 $N^2 - OB$ 上的一个广义自由区划分, 简称广义划分, 记所有广义划分的集合为 $GP(N^2 - OB)$. 若 $P_g, \tilde{P}_g \in GP(N^2 - OB)$, 使得任意 $\tilde{F}_g \in \tilde{P}_g$, 存在 $P' \subseteq P_g$, 使得 $\tilde{F}_g = \bigcup_{F_g \in P'} F_g$, 则称 \tilde{P}_g 包含 P_g , 也就是说, \tilde{P}_g 中的任意一个自由区都可被 P_g 中的自由区所覆盖, 记为 $P_g \prec \tilde{P}_g$. 对于 P_g , 若不存在 \tilde{P}_g , 使得 $P_g \prec \tilde{P}_g$, 则称 P_g 为极大广义划分, 简称极大划分. 若不特别注明, 以下的广义划分均为极大划分.

显然, 由连接图 G_C 和 G_T 上的顶点和边将平面划分成的矩形区域的集合为广义划分(但不是极大划分), 但

对 G_F 则不然.对基于广义划分的连接图而言,尚需要更多的约束条件,故定义如下:

定义 10(正规广义自由区划分). 对于广义划分 $P_g \in GP(N^2 - OB)$,若 $F_{gi} \in P_g$ 满足以下条件:

- (1) 自由区的边界就在障碍边的轨迹上: $bound(F_{gi}) \subset \bigcup_{ob_i \in OB} (T(ob_i))$;
- (2) 自由区的极边 $ex(F_{gi})$ 与障碍区的极边交叠或者包含一个极顶;
- (3) 任意 $F_{g,i}, F_{g,j} \in P_g$,若 $F_{g,i} \cap F_{g,j}$ 不为空,则所有的交点在一条公共边上;
- (4) 任意 $F_{g,i}, F_{g,j} \in P_g$,其边界上的点的距离大于 1 或者为 0,

则称 P_g 为 $N^2 - OB$ 上的正规广义自由区划分,简称正规划分,记其集合为 $UP(N^2 - OB)$.同样定义极大正规划分(可以看到,图 G_C 和 G_T 产生的广义划分都不是正规划分,更不是极大划分).

在定义中,条件(1)保证,若两点之间的最短路径长度大于两点之间的距离,则最短路径可以用正规划分中的凸多边形的边界来构成.条件(2)使得“悬空”的凸多边形不会出现,且保证了正规划分中凸多边形的数量为极边数量的常数倍(如定理 5).条件(3)使得处在不同自由区中的同一个顶点之间通过边界的连接惟一,从而使得在求解最短路径过程中使用的规范路径(将在第 4 节中定义)的概念成立.条件(4)保证,如果两个区域在平面上是相邻的,则其交集非空,这是为下面描述方便而设置的.平凡的正规划分可以通过平面扫描的方法得到(具体算法见下一节),从而使正规划分的存在性得到保证,故可得到关于连接图的定义如下:

定义 11(广义连接图). 对于一个正规划分 $P_g \in UP(N^2 - OB)$,定义图 $G_G = (V, E)$,其中, $|V| = |P_g|$, 顶点 $v \in V$ 一一对应到自由区 $F_g(v) \in P_g$,且 $(u, v) \in E \Leftrightarrow F_g(v) \cap F_g(u) \neq \emptyset$, 则称 G_G 为 P_g 所定义的广义连接图,记为 $G_G = G(P_g)$.

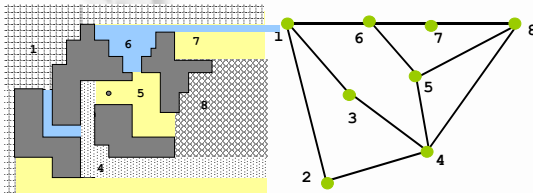


Fig.2 An example of maximum formal partitioning and corresponding generalized connection graph
图 2 极大正规划分的例子和对应的广义连接图

连接图 $G_G = G(P_g)$ 的顶点数 $|V| = \Theta(t)$,边数 $|E| = \Theta(t)$,其中 t 为 OB 中的极边的数量.

证明:证明分两部分,分别得到线性的上下界.

[上界]:对任意极顶 u ,不妨设它是上极边的右极顶(其他情况可通过旋转和镜面变换得到同样的位置而不改变拓扑结构),以 u 为原点构造坐标平面,可以看出,一个象限中至多有一个广义自由区的边界包含该极顶,故包含该顶点的广义自由区至多 3 个(另外一个象限为障碍区).由正规划分定义的条件(2)可知,任何一个广义自由区必然与一个极顶相邻,因此,自由区数 $|V| = |P_g| \leq \sum_{v \text{ 为极顶}} \text{与} v \text{ 相邻之广义自由区数} \leq 3 * \text{极顶数} \leq 6t$.由定理 4 可知, G_G 为平面图,故有 $|E| \leq 3|V| - 6 < 18t$.

[下界]:对障碍的个数 n 用归纳法来证明: $|V| \geq n$.当 $n=1$ 时, $|V| \geq 2$,命题成立.假设当 $n=k$ 时命题成立.当 $n=k+1$ 时,对其正规划分 P_g ,任意取出一个障碍,包含该障碍区极顶的自由区凸多边形至少有两个,根据定义 10 的条件(3),可将这些自由区凸多边形合并后重新组合为凸多边形,显然,其数量至少减 1;其余的凸多边形不变,得到新的正规划分 P'_g ,由归纳假设 $|P'_g| \geq k$,得到 $|P_g| \geq |P'_g| + 1 > k + 1$.归纳成立.故 $|V| \geq n = t/4$.命题得证. \square

G_G 与一般连接图的不同之处在于, G_G 的顶点表示一段子路径而不是边;同时 G_G 不是惟一的.因此存在最优化问题:求 $P_g \in UP(N^2 - OB)$,使得 $G_G = G(P_g)$ 的顶点数最少.这是 NP 难题,无法在多项式时间内得到最优解.故我们从不同的出发点得到近似最优构造,由定理 5,显然有合法的正规划分均为对最优解的线性近似.

3 极大正规划分的构造

我们用扫描线算法直接构造极大正规划分.在平面上记录所有障碍的极顶,其 x 坐标的增序排列为

x_1, x_2, \dots, x_l ($x_i < x_{i+1}, 1 \leq i \leq l-1$), 任取 x_0, x_{l+1} , 使得 $x_0 < x_1, x_l < x_{l+1}$. 设第 i 根扫描线 l_i 为 $x=x_i$. S_i 为在扫描线 l_i 上的扫描区间集合, 其中每个扫描区间表示为 $\langle u, d \rangle$, u, d 分别为其上端和下端, 同时用标记 up_ep 和 $down_ep$ 分别表示扫描区间的上端和下端是否能继续向上和向下扩展. 设置初始扫描区间 ($t=0$) $S_0 = \{(+\infty, -\infty)\}$ 及标记均为真. 对于图 1 的实例, 极大正规划分的构造如图 3 所示.

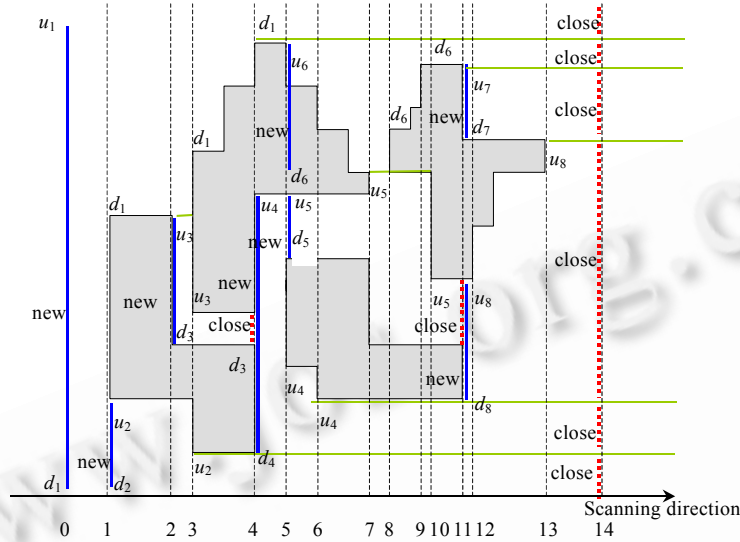


Fig.3 Construction of maximum formal partitioning using plane scanning

图 3 扫描线方法构造极大正规划分的示意图

图中扫描方向为从左到右, 每个扫描线的标号在图底部显示; new 表示创建一个新的扫描区间, close 表示关闭一个扫描区间; 同时标明当从左到右扫过某一区域时对区间上下界的更新 (u_i, d_i). 对于第 t 个扫描线 l_t , 我们有如下分析:

(1) 若在 l_t 上包含障碍的左极边 $e = \langle (x_t, y_1), (x_t, y_2) \rangle (y_1 < y_2)$, 则有 3 种情况:

a) 扫描区间 $s = \langle u, d \rangle$ 被完全包含在该边内, 即 $y_1 \leq d < u \leq y_2$, 则关闭扫描区间: $S_t = S_t - \{s\}$;

b) 扫描区间与该边相交, 不妨设 $y_1 \leq d < y_2 \leq u$ (两个等号不能同时成立), 则裁减扫描区间: 置 $s = \langle u, y_2 \rangle$, $down_ep(s) = FALSE$, 反之亦然, 如图 3 中扫描线 3 和扫描线 8 所示;

c) 该边被完全包含在扫描区间 $s = \langle u, d \rangle$ 内, 即 $d < y_1 < y_2 < u$, 则分裂扫描区间: 随机选择上端或下端继续扩展, 而在另一端生成新的扫描区间. 不妨设扩展上端, 如图 3 中扫描线 1 所示, 新建一个扫描区间 $s' = \langle u', d' \rangle$, 其中 $d' = d, u' = y_1, S_t = S_t \cup \{s'\}$, 同时设置 $s = \langle u, y_2 \rangle$. 对扩展下端的情况作完全类似的处理, 如图 3 中的扫描线 5.

(2) 若在 l_t 上包含障碍的右极边 $e = \langle (x_t, y_1), (x_t, y_2) \rangle (y_1 < y_2)$, 则根据前面的算法过程, 此时必有两个扫描区间 $s_1 = \langle y_1, d \rangle, s_2 = \langle u, y_2 \rangle$, 分以下 3 种情况来处理:

a) 若 $up_ep(s_1) = FALSE, down_ep(s_2) = FALSE$, 则合并这两个扫描区间, 并设置新的区间为 $s = \langle u, d \rangle$, $S_t = S_t - \{s_1, s_2\} \cup \{s\}$;

b) 否则, 若已经形成的两个自由区有过公共边 (防止最终产生的划分中的两个区域有两条公共边), 则关闭区间 s_2 , 生成新的扫描区间 $s_3 = \langle u, y_1 \rangle$, 或者关闭区间 s_1 , 生成新的扫描区间 $s_3 = \langle y_2, d \rangle$;

c) 否则 (不会产生两条公共边的情况), 若 $up_ep(s_1) = FALSE$, 则设置 $s_2 = \langle u, y_1 \rangle$, 若 $down_ep(s_2) = FALSE$, 则设置 $s_1 = \langle y_2, d \rangle$, 否则, 可随机选择以上方案之一. 这种情况的实例如图 3 中扫描线 4、7、11、13 所示.

(3) 若在 l_t 上包含障碍的下极边上的顶点 $v = (x_t, y)$, 且该顶点不在某条左(右)极边上, 有 $s_i = \langle u, d \rangle \in S_t$ 使得 $y \in [d, u]$, 则有两种情况:

a) 若该顶点为左极顶, 则置 $s_i = \langle y, d \rangle, up_ep(s_i) = FALSE$, 如图 3 中扫描线 3、6、10 所示, 即为保持生成的区域为凸的, 不允许再次将区间的上界向上扩展, 否则将产生凹的极边;

b) 若该顶点为右极顶,则必有 $y = u$ (在左极顶处设置), 设 y' 为 v 所在的障碍的右极边的下顶点的 y 坐标, 若 $up_ep(s_i) = \text{FALSE}$, 则新建一个扫描区间 $s' = \langle u', d' \rangle$, 其中 $d' = y$, $u' = y'$, $S_i = S_i \cup \{s'\}$, 如图 3 中扫描线 12 所示; 否则, 置 $s_i = \langle y', d \rangle$. 注意到, 这样生成的扫描线包含了障碍中的点, 但根据空极区的定义, 在生成多边形和求最短路径时是正确的.

(4) 若在 l_i 上包含障碍的上极边上的顶点, 且该顶点不在某条左(右)极边上, 则如与(3)的处理方法完全类似, 左极顶使得 d_i 增大, 如图 3 中扫描线 4、9 所示; 右极顶则使得 d_i 减小, 如图 3 中扫描线 7 所示, 或者产生一个新的扫描区间, 如图 3 中扫描线 2、5、12 所示.

在最后一扫描线上关闭所有的扫描区间, 则显然每个扫描区间构成一个广义自由区, 其边界由扫描区的上下界的变化轨迹得到, 两者之间是一一对应的.

从扫描线算法构造扫描区间的规则, 显然有如下定理:

定理 6. 极大正规划分的扫描线算法所得到的平面划分为极大正规划分, 且划分的多边形最多为 t 个.

定理 7. 极大正规划分的扫描线算法的时间复杂度为 $O(t \log t)$.

对定理 7, 只需注意到扫描线上需要处理的极边和极点的数量为 $O(t)$, 扫描区间的数量为 $O(t)$, 故对于每条极边只需要 $\log t$ 时间得到对应的扫描区间, 注意到对扫描区间的处理总可以在常数时间内完成, 故得到扫描区间需要 $O(t \log t)$ 时间. 扫描区间集合与极大正规划分的对应使得总的扫描区间数为 $O(t)$, 其总上下界的变化次数为 $O(t)$ 次, 故从扫描区间得到的极大正规划分只需 $O(t)$ 时间, 从而整个算法的时间复杂度仅为 $O(t \log t)$. 显然, 空间复杂度仅为 $O(t)$.

在完成极大正规划分的构造以后, G_G 的构造是很自然的, 此处不再论述.

4 用 G_G 求两点间的最短路径

对于 $N^2 - OB$ 中任意两点, 它们间的最短路径的判定有两种情况:

(1) 若两点处于同一广义自由区, 则此两点间有不带极边的最短路径, 其长度等于两点间的距离;

(2) 否则, 两点所定义的矩形区中有障碍存在(作更进一步的优化则可判定两点所定义的矩形区域中是否有横跨的障碍, 如果没有, 则通过扫描算法直接得到), 且它们不在同一广义自由区中, 则最短路径需跨越广义自由区, 那么, 最短路径和所跨越的广义自由区的关系如何?

为此, 在 G_G 中, 我们引入一种特殊的路径的定义如下:

定义 12(规范路径). 对于一个正规划分 $P_g = \{F_g\} \in UG(N^2 - OB)$, 两节点 $s, t \in N^2 - OB$, 但分属不同的广义自由区, 定义这两个广义自由区在广义连接图中的路径为 $SF = \{F_{g,i} \mid 0 \leq i \leq k, s \in F_{g,0}, t \in F_{g,k}\}$, 其中 $F_{g,i}$ 与 $F_{g,i+1}$ 相邻, 且 $\forall i \neq j, F_{g,i} \neq F_{g,j}$, 如果路径 $P = p_0 p_1 \dots p_{k+1}$ 满足条件:

$$p_0 = s, p_{k+1} = t, p_i \text{ 在 } F_{g,i} \text{ 的边界上, 并且满足 } \rho(p_i p_{i+1}) = \min_{\tilde{p} \in F_{g,i+1}} \rho(p_i \tilde{p}), 0 \leq i < k,$$

则称 P 为正规划分 P_g 中点 s 到 t 在 SF 上的规范路径.

如图 4 所示, 规范路径的含义是在路径所经过的每个广义自由区中的子路径都是最短的. 在正规划分的约束条件下, 条件(3)要求相邻的广义自由区只有一条公共边, 这样才能保证局部最优是全局最优的. 如图 4 所示, 当条件(3)不满足时, 两广义自由区相邻的方向不只有一个, 局部最优路径选择了一个方向, 而当这个方向和下一个广义自由区的邻接方向不一致时, 它不是最优的, 即在条件(3)不成立时, 规范路径的定义需基于不同方向上的局部最优路径的组合, 而不再是惟一的, 这样虽然可以轻微地减少广义自由区的数目, 但增加了不同方向选择时的空间和时间开销, 并增大了算法实现的复杂性, 是不可取的.

在正规划分后的广义连接图 G_G 中求两点间的最短路径是由下面的定理保证的.

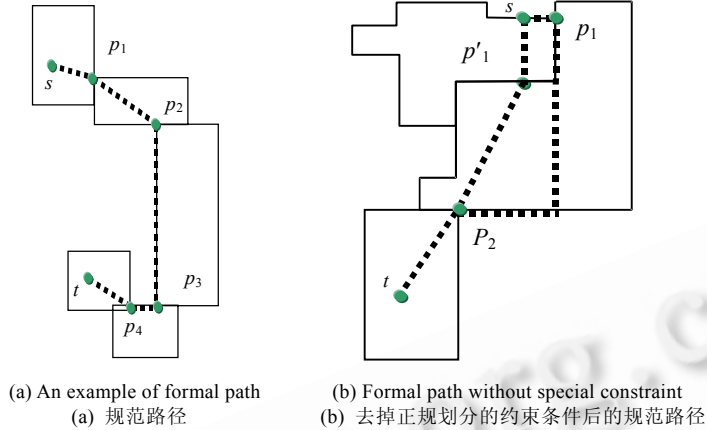


Fig 4
图 4

定理 9(规范路径的最优性). P 为正规划分 P_g 中点 s 到 t 在 SF 上的规范路径,则在点 s 到 t 的所有只经过 SF 中自由区的路径中,此规范路径的长度最小:

由于 P_g 是将 $R^2 - OB$ 划分为 $|P_g|=k$ 个凸区域, $|E(G_G)|=e$ 条边的平面划分,采用 Kirkpatrick 算法,每个点可在 $O(\log e)$ 时间内定位于其中的某个区域内(空间复杂度为 $O(k+e)$,预处理时间为 $O(k+e)$).

在得到点 s 所在的广义自由区以后,根据 G_G 得到广义自由区的相邻关系,构成一个状态空间,寻找两极顶 s, t 间的最短路径是在这个状态空间的搜索问题,而与一般的状态空间搜索不同的是:两节点间的边的权值是在搜索到这条边时才能确定的.类似于 G_F ,我们用带启发策略的 A^* 算法求解,详细的算法见文献[8,9].

5 实验结果

表 1 是图 1 中的实例用各种连接图表示时的顶数和边数,可以明显地看到其优劣性.表 2 总结了关于 3 种连接图和广义连接图的复杂度的结论,注意到,在图中求最短路径的复杂度等同于图的边数的复杂度,故在表中没有明确列出.

Table 1 Comparison among different connection graphs for the same instance

表 1 对于同一个实例的不同连接图的比较

	G_G	G_G	G_G	G_G
Number of vertices	120	46	26	8
Number of edges	154	64	35	11

Table 2 Complexity comparison among four kinds of connection graphs

表 2 4 种连接图的复杂度对比

	G_C	G_F	G_F	G_G
Number of vertices	$O(e^2)$	$O(r^2)$	$O(t)$	$\mathcal{O}(t)$
Number of edges	$O(e^2)$	$O(r^2)$	$O(t \log t)$	$\mathcal{O}(t)$
Maximum degree of vertices	$O(1)$	$O(1)$	$O(t)$	Planar graph
Construction complexity	$O(e^2 \log e)$	$O(r^2 \log t)$	$O(t \log t)$	$O(t \log t)$

我们在 PII333 上实现了产生图的算法和在图上求最短路径的算法,对其性能进行了比较,其结果如图 5 所示(图中横坐标为随机生成的障碍区的数目).无论从理论上还是实际上, G_G 均优于现有的几种方法.

6 结 论

本文提出了广义自由区、广义连接图和极大正规划分的概念,并在此基础上给出了构造广义连接图 G_G 的方法.广义连接图用来表征广义自由区之间的邻接情况,此图较以往的连接图更为简单,使得基于连接图搜索的

最短路径算法的时间复杂度由 $O(t \log t)$ 降低到 $\Theta(t)$.

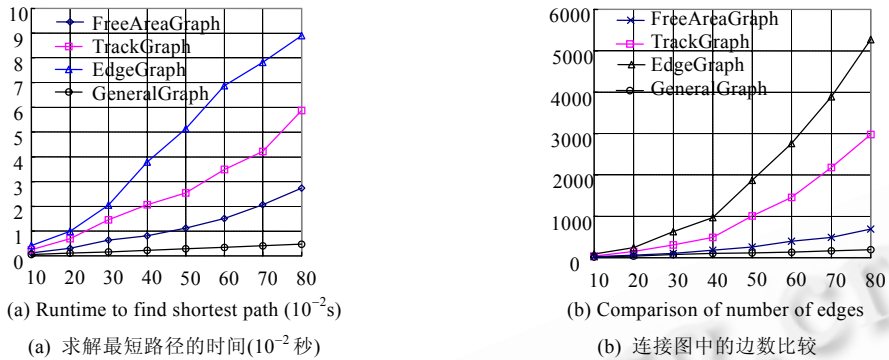


Fig.5 Comparison among the four typical connection graphs

图 5 4 种典型的连接图的性能比较

References:

- [1] Lee CY. An algorithm for path connections and its applications. IRE Transactions on Electronic Computers, 1961, EC-10:346~365.
- [2] Akers SB. A modification of Lee's path connection algorithm. IEEE Transactions on Electronic Computers, 1967, EC-16:97~98.
- [3] Soukup J. Fast maze router. In: Proceedings of the 15th Design Automation Conference. Las Vegas: ACM Press, 1978. 100~102.
- [4] Hightower DW. A solution to line routing problems on the continuous plan. In: Proceedings of the 6th Annual Design Automation Workshop. ACM Press, 1969. 1~24.
- [5] Clarkson KL, Kapoor S, Vaidya PM. Rectilinear shortest path through polygonal obstacles in $O(n \log^2 n)$ time. In: Proceedings of the 3rd Annual Symposium on Computational Geometry. New York: ACM Press, 1987. 251~257.
- [6] Zheng SQ, Lim JS, Iyengar SS. Finding obstacle-avoiding shortest paths using implicit connection graphs. IEEE Transactions on Computer Aided Design, 1996, 15(1):103~110.
- [7] Wu YF, Widmayer P, Schlag MDF, Wong CK. Rectilinear shortest paths and minimum spanning trees in the presence of rectilinear obstacles. IEEE Transactions on Computers, 1987, 36(3):321~331.
- [8] Zhou Z. Obstacle-Avoiding minimum rectilinear steiner tree problem [MS. Thesis]. Hefei: University of Science and Technology of China, 1998 (in Chinese with English Abstract).
- [9] Zhou Z, Chen GL, Gu J. Finding obstacle-avoiding shortest path using connection graph with $O(t \log t)$ edges. Chinese Journal of Computers, 1999, 22(5):519~524 (in Chinese with English Abstract).

附中文参考文献:

- [8] 周智. 有障碍的 Manhattan 空间中的最小 Steiner 树问题 [硕士学位论文]. 合肥: 中国科学技术大学, 1998.
- [9] 周智, 陈国良, 顾钧. 用 $O(t \log t)$ 的连接图求有障碍时的最短路径. 计算机学报, 1999, 22(5):519~524.