



清华大学 自动化系

Department of Automation, Tsinghua University

智能机器人运动规划

Motion Planning

张涛

清华大学自动化系

1. 导航能力：规划和反应

2. 路径规划

2.1 图搜索方法

2.2 势场路径规划

3. 避障

4. 小结

1. 导航能力：规划和反应



互
补

规划：指导整个机器人的行为以达到它的长远目标

反应：在执行期间，机器人必须对一些不可预见的事件(如障碍)以某种方式做出反应，使之仍然达到目标。如果没有反应，规划的结果将是徒劳的，因为机器人在物理上将永远不会达到目标。

路径规划(path planning)和**避碰**是实现移动机器人导航需要两种互补能力

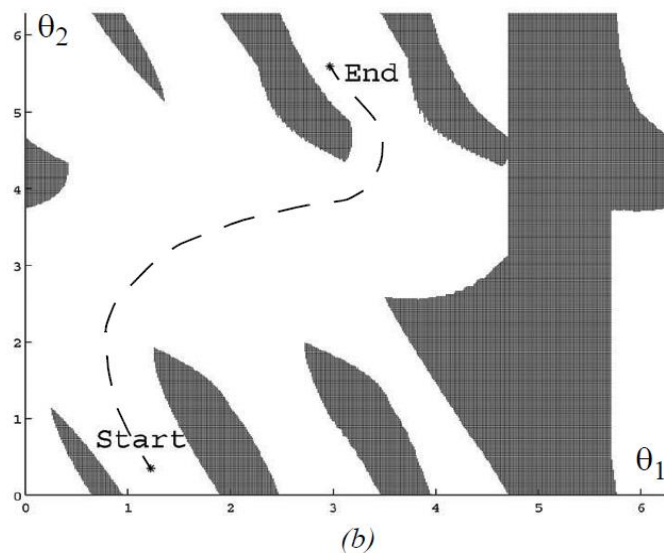
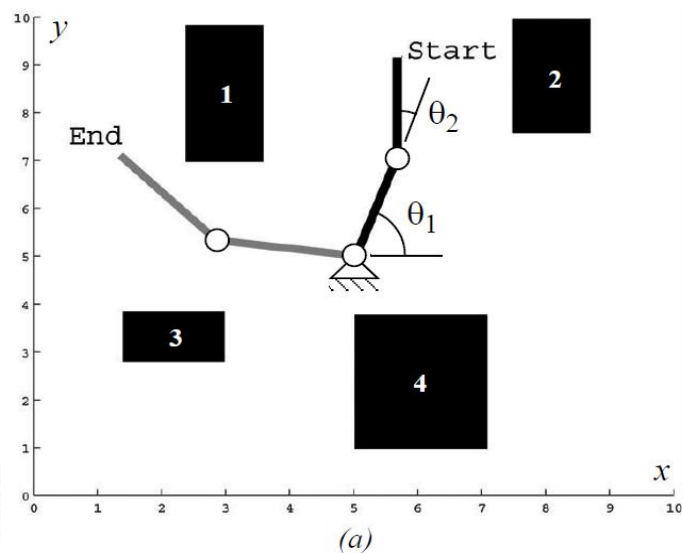
完备性(completeness)：一个机器人系统是完备的，当且仅当对于所有可能的情况(例如，初始置信状态、地图和目标)，当存在一条到达目标置信状态的轨迹时，系统将获得目标置信状态。

- 6DOF的机械手的路径规划问题远比在平坦地面移动的移动机器人复杂。由于自由度大幅度降低，移动机器人所用的路径规划算法相对简单。
- 由于生产效率的要求，工业机械臂通常以尽可能快的速度运行，非线性的动力学和运动学，使路径规划与执行复杂化。相反，移动机器人的运行速度较低，在路径规划时极少考虑动力学，这进一步简化了移动机器人路径规划问题。

2. 路径规划



位形空间(*Configuration space*): 假设机器人有 k 个自由度, 则机器人的每个状态或位形, 可用 k 个实数 q_1, \dots, q_k 表示。 $[q_1, \dots, q_k]^T$ 表示位形空间 C 中的一个点 p 。



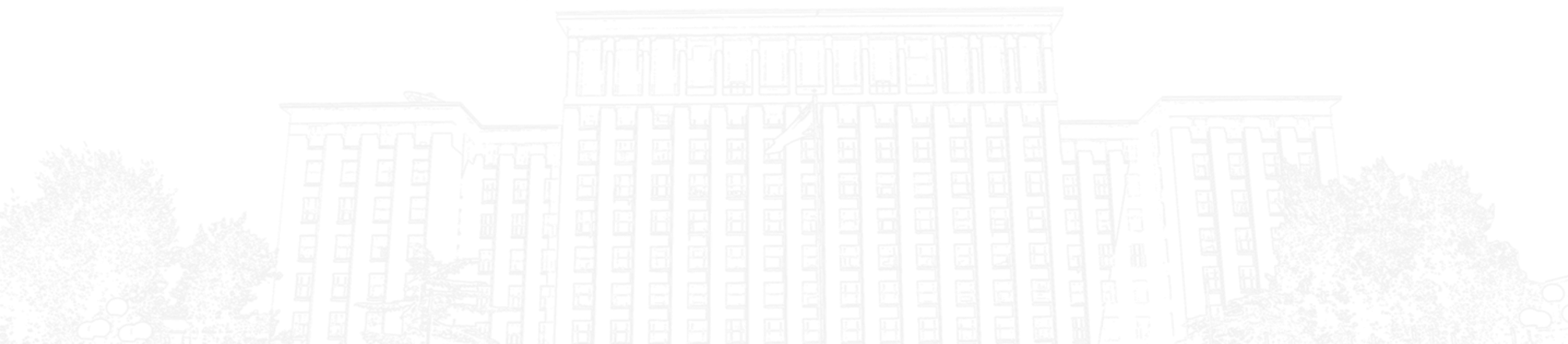
左图表示两连杆平面机械臂的物理空间和位形空间

(a) 在物理空间中的任务定义,
(b) 对应的位形空间, 给出了
自由空间和可行路径

2. 路径规划



- 路径规划的目标是在**物理空间**中找到一条从初始点位置到目标点位置的路径，避免与所有障碍物碰撞。
- 在**位形空间**中，描述路径规划问题更直接。如果将位形空间障碍物子空间定义为 O ，则机器人可安全移动的自由空间为： $F=C-O$ 。



2. 路径规划



- 对于在平地上移动的机器人，一般可用3个变量表示机器人位姿 (x, y, θ) 。但多数轮式移动机器人采用差速驱动(differential drive)，非完整约束限制了在 (x, y, θ) 处的速度，这类路径规划问题求解更复杂。
- 在研究移动机器人学的路径规划问题时，通常假设机器人是完整的(holonomic)，并进一步将机器人**看作一个点**。因此，位形空间简化为只有 x 和 y 轴的 2D 表示。
- 将机器人简化为一个点，必须将各个障碍物按机器人的半径尺寸进行膨胀，这样得到简化的结构空间描述。

路径规划方法分类 机器人的环境表示可以从连续的几何描述到基于分解的几何图，甚至拓扑图。因此，路径规划问题的第一步都是将连续的环境模型转换成适合于选择路径规划算法的离散模型。根据离散分解的方式不同，路径规划方法可分为两类：

- **图搜索** (Graph search): 首先构造自由空间中的连通图，然后进行搜索。图的构造通常离线执行。
- **势场规划** (Potential field planning): 将数学函数直接作用于自由空间, 跟随这个函数的梯度可以到达目标。

图搜索过程主要分为两个步骤：

- (1) 图的构建：节点放在哪，节点间如何通过边连接
- (2) 图搜索：寻找最优解（路径）

2.1.1 图的构建

先将环境模型表示为自由空间 F 和占用空间 O ，再将这一表示分解为可用某种算法搜索的图。

挑战：如何构建一组节点和边，使机器人能够在其描述的自由空间的任何地方移动，同时需限制图的总规模。

2.1 图搜索方法



2.1.1 图的构建

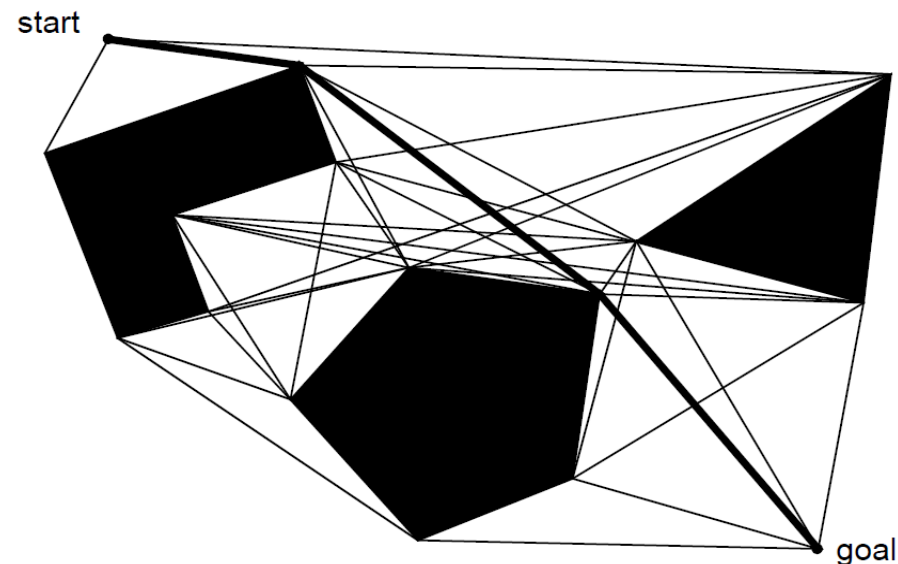
路线图法 (road map approach)

(1) 可视图 (Visibility graph)

- 机器人用点来描述
- 障碍物用多边形描述

多边形位形空间的可视图，由连接彼此可见的全部**顶点对**的边组成(初始和目标位置也被看作顶点)。显然，用无遮挡的直线连接这些顶点距离最短。

路径规划的任务就是沿着可视图定义的路径，寻找从初始位置到目标位置的最短路径。



可视图示例

2.1 图搜索方法

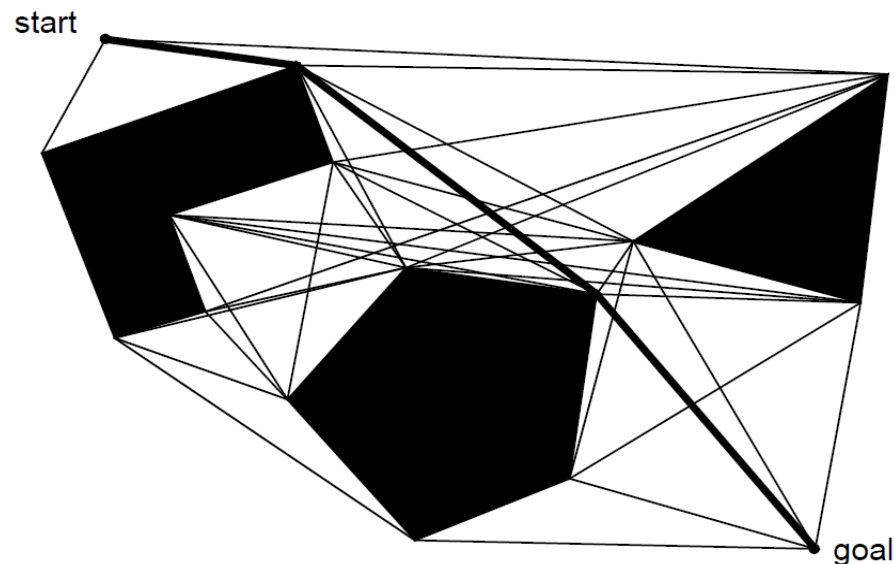


(1) 可视图 (Visibility graph)

优势:

- 简单, 直观
- 可以证明可视图规划的路径是最短的
(机器人尽可能地接近障碍物)

如果要求机器人与障碍物保持合理的距离, 则需要修正, 会牺牲可视图路径规划的长度最优性。



可视图示例

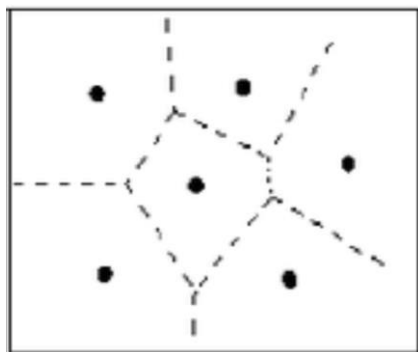
可视图法用于稀疏环境快速、有效，该方法适用于障碍物稠密的环境吗？

- ☐ A 同样适用
- ☐ B 不适用，速度慢
- ☐ C 不适用，可能无效

提交

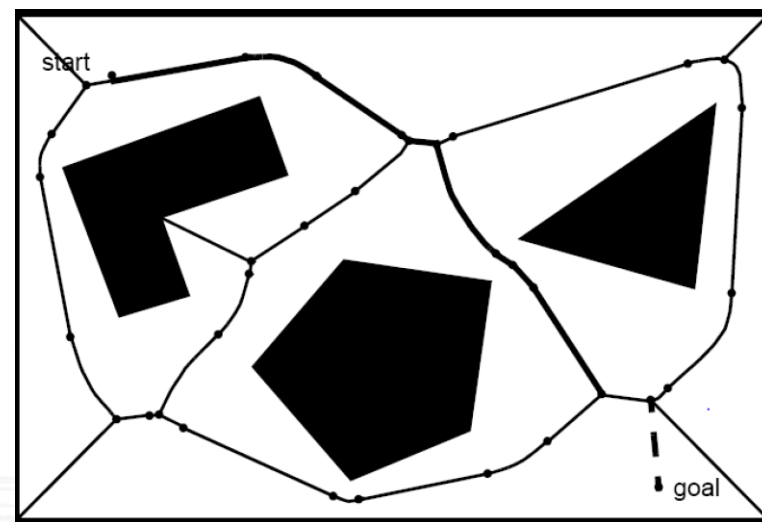
(2) 沃罗诺伊图 (Voronoi diagram)

Voronoi 图由一组连接两相邻点直线的**垂直平分线**组成的连续多边形组成



Voronoi图示意

用于平面多边形
障碍位形空间的
路线图构建



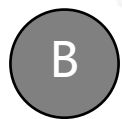
- Voronoi 图由线段组成，线段上的所有点到最近多个障碍物的距离均相等。
- Voronoi 图最大化机器人与障碍物之间的距离。
- 当结构空间障碍物是多边形时，Voronoi 图由直线段和抛物线段(到直线和点之间保持最小距离)组成。

Voronoi 图法在机器人传感器检测范围有限时显然达不到预期结果。但配备有距离传感器(如激光或超声测距传感器)的机器人可以使用简单的控制规则实现：使其传感器值的局部极小值最大化，即可实现对规划轨迹的跟踪，因此可减少车轮编码器精度的依赖。

你认可上述分析吗？



认可



不认可

提交

2.1.1 图的构建

路线图法 (road map approach)

(3) 单元分解法

将位形空间**分解**为单元，把单元分为自由的和被占用的。基本的单元分解路径规划思路如下：

- 将自由空间分解为“单元”的连接区。
- 确定哪些开放单元是相邻的，并构建一个“连接图”。
- 找出初始位形和目标位形所在的单元，并在连接图中搜索一条连接初始和目标单元的路径。

单元之间的边界：精确 or 近似

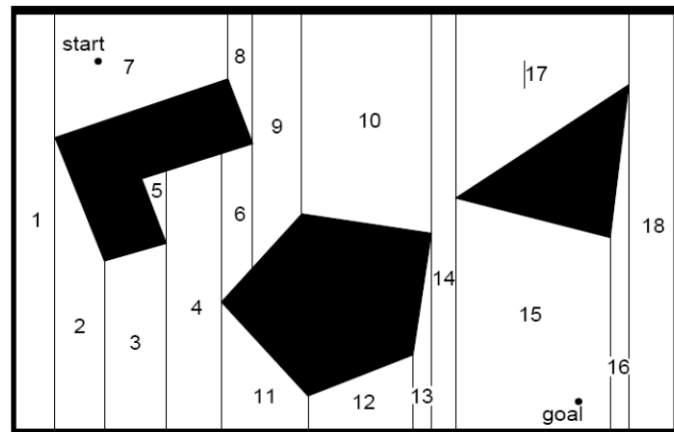
2.1 图搜索方法



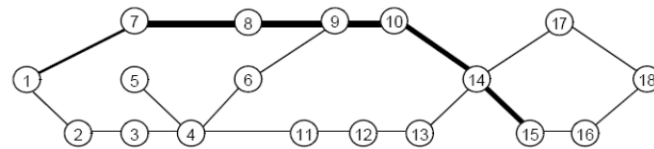
(3) 单元分解法

1) 精确的单元分解

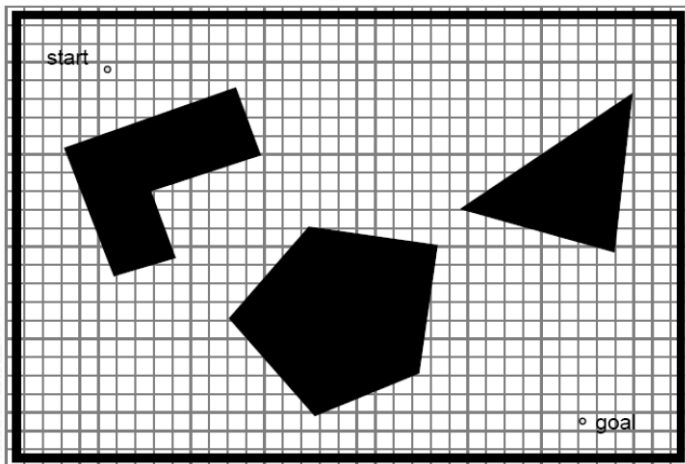
由于实现复杂，实际较少使用



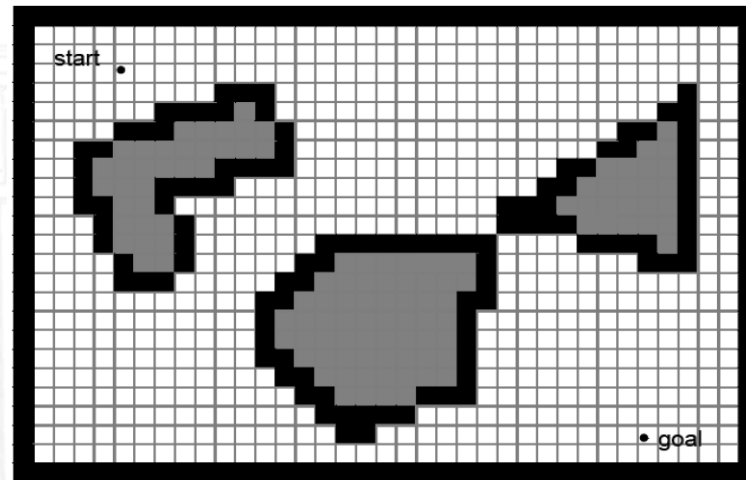
精确的单元分解示例



2) 近似的单元分解——栅格图



固定尺度
单元分解



变尺度
单元分解

2.1.2 图的搜索

假设环境地图已转化为连通图，路径规划的目标是在地图的**连通图**中找到从起点到目标点的最优路径，优化指标可以是最短路径等。

为讨论方便，用 $g(n)$ 表示从起始节点到任意给定节点 n 的累计代价函数，从节点 n 到相邻节点 n' 的代价为 $c(n, n')$ ，从节点 n 到目标节点的期望代价（启发代价）表示为 $h(n)$ ，则从起点、经过节点 n 到目标的总的期望代价为：

$$f(n) = g(n) + \varepsilon \cdot h(n)$$

其中 ε 为与算法有关的参数。

2.1.2 图的搜索

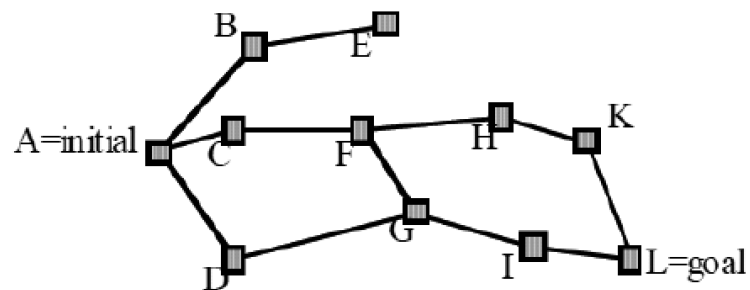
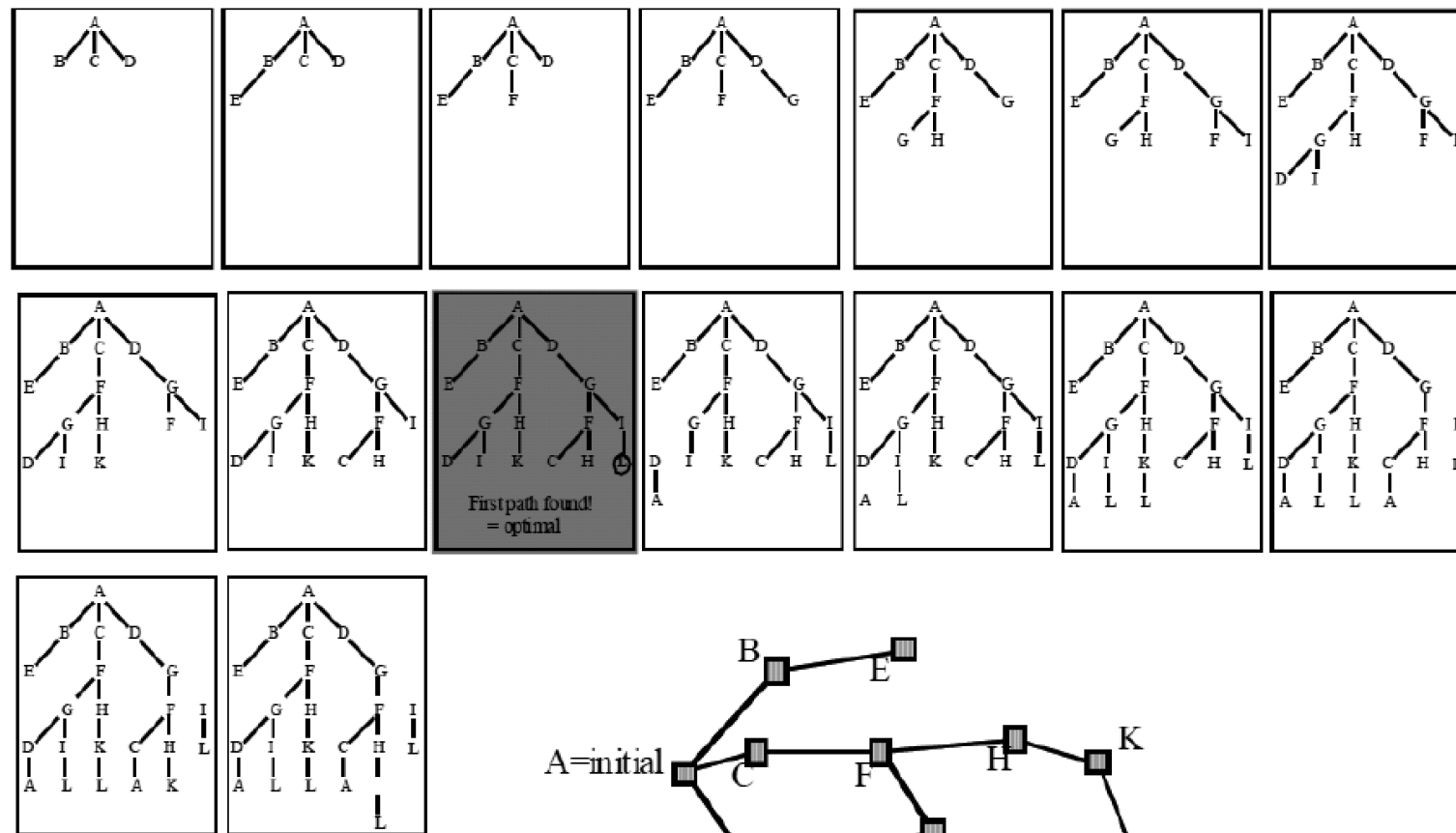
$$f(n) = g(n) + \varepsilon \cdot h(n)$$

- $\varepsilon=0$ 如果图中每条边的遍历成本相同，可采用深度优先或广度优先搜索；若成本非均匀，可采用Dijkstra的算法和其变体，当然计算复杂性相对更高
- $\varepsilon \neq 0$ 意味着使用启发函数，其包含了关于问题的附加信息，因此可能使搜索更快地收敛。如A*算法等。

2.1 图搜索方法

2.1.2 图的搜索

- 广度优先搜索



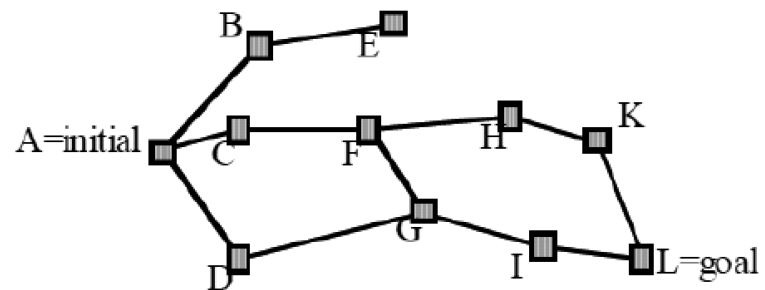
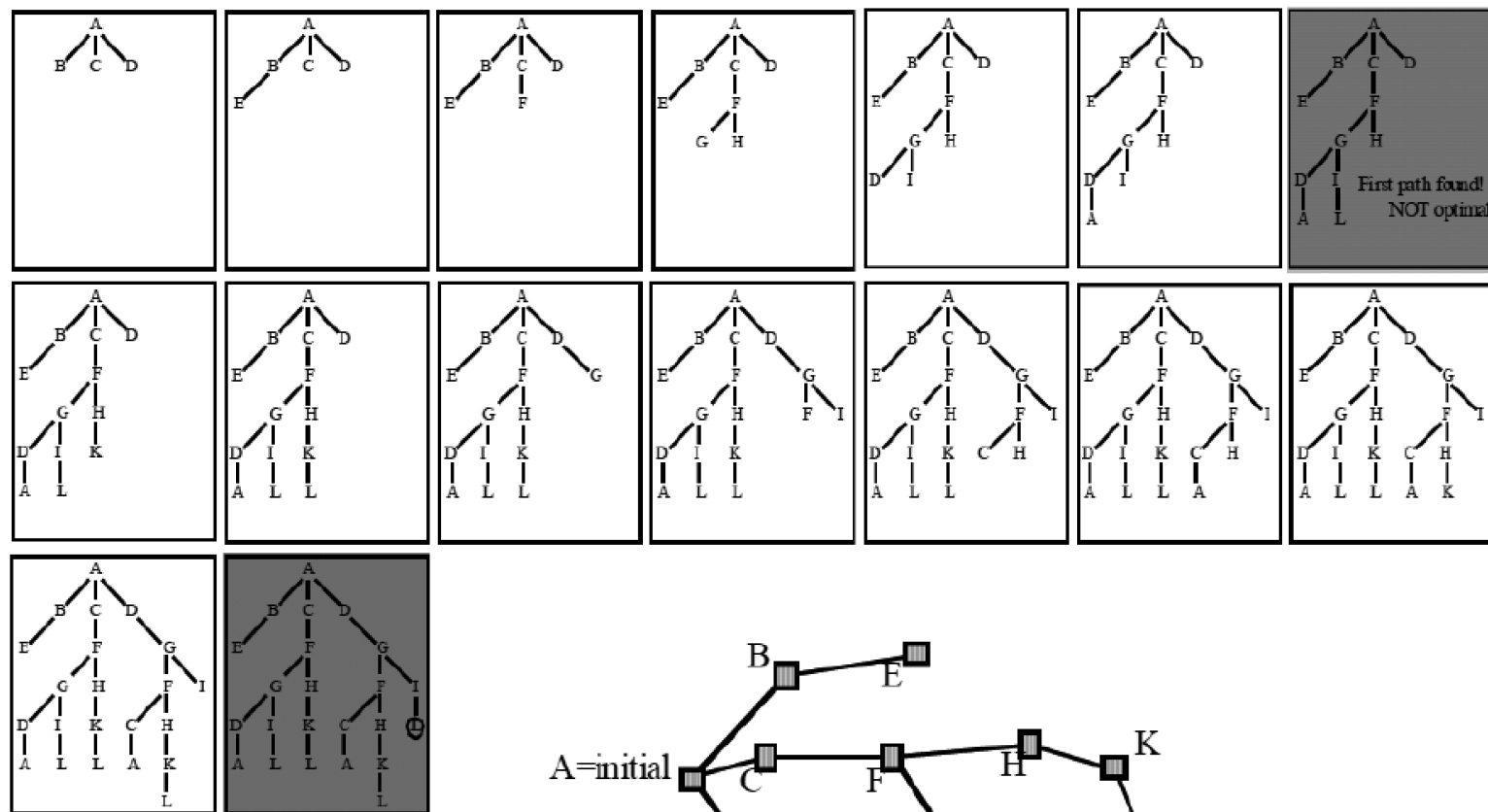
广度优先搜索示例

2.1 图搜索方法



2.1.2 图的搜索

- 深度优先搜索



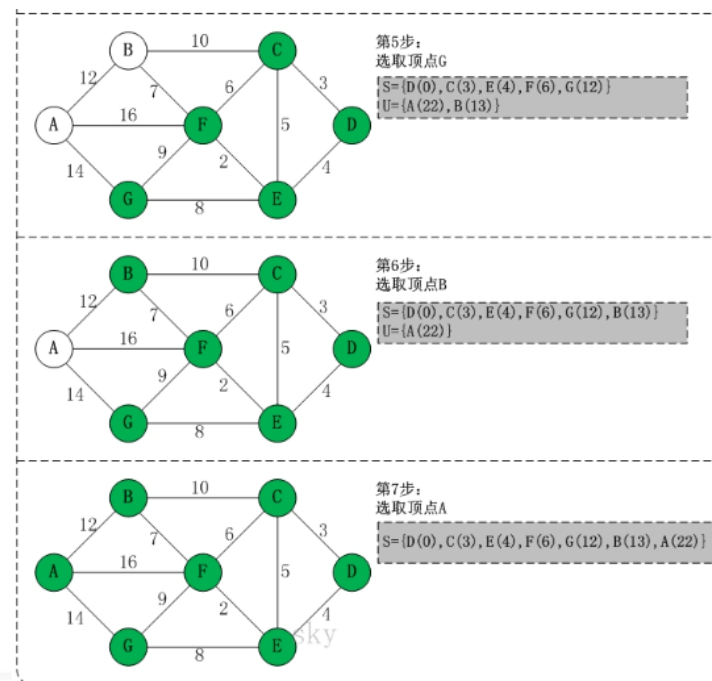
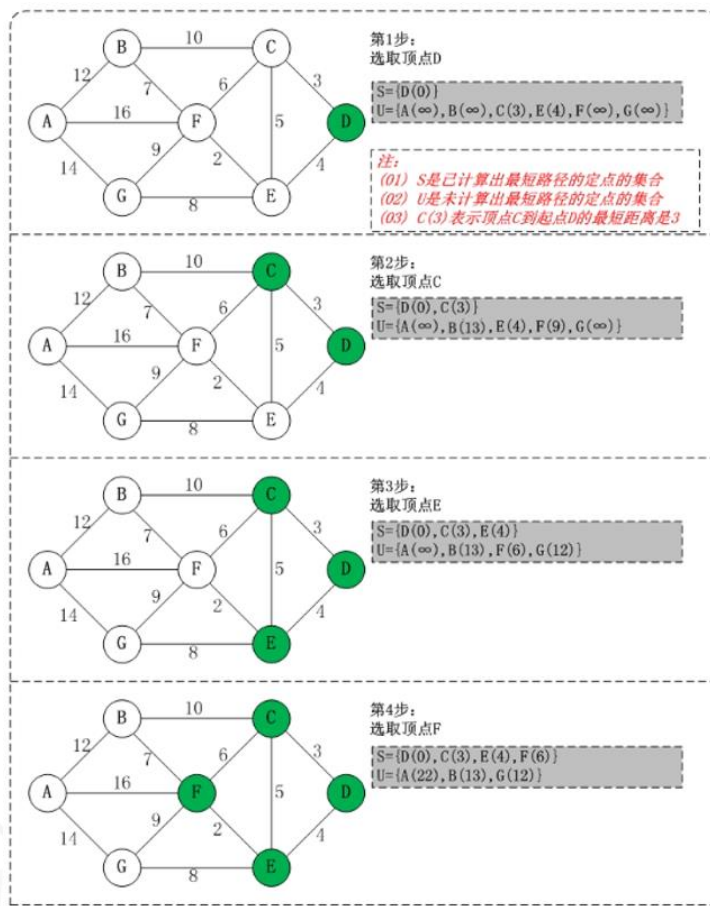
深度优先搜索示例

2.1 图搜索方法

2.1.2 图的搜索

● Dijkstra算法

与广度优先类似，
但边代价可以是
任意正值，可保
证得到最优解。



Dijkstra算法示例

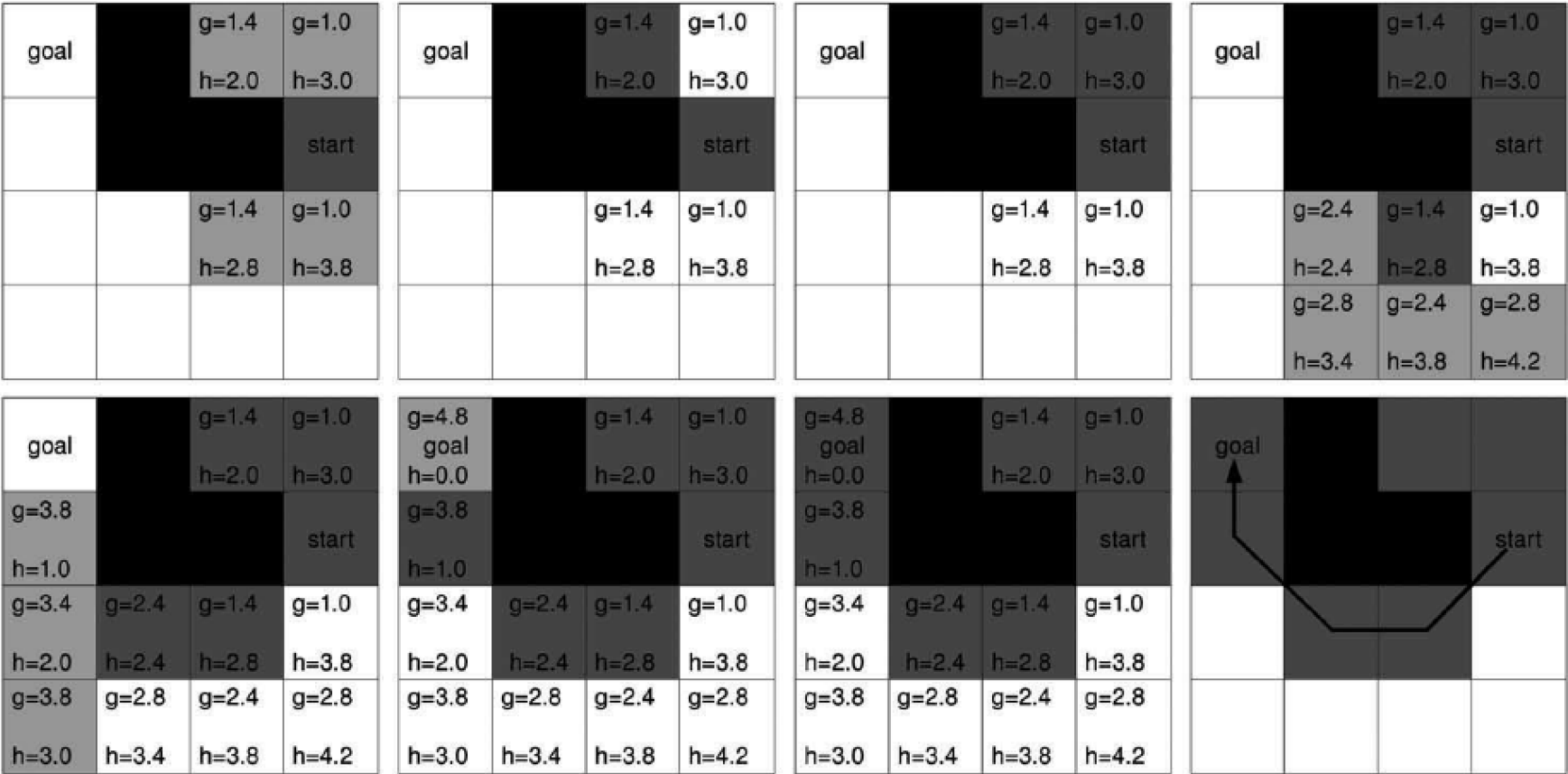
From <https://blog.csdn.net/heroacool/article/details/51014824>

在机器人应用中，通常从目标位置开始搜索，向外层层扩展，直到扩展到起点。
这样可以得到从图中任一节点到目标节点的最低代价路径，避免重复规划。

2.1.2 图的搜索

● A*搜索 (启发式搜索)

$$f(n) = g(n) + \varepsilon \cdot h(n) \quad \text{where } \varepsilon=1$$



A*算法示例：障碍单元为黑色，被扩展单元为深灰色，当前步可扩展单元为灰色

[1] Roland Siegwart, Introduction to autonomous mobile robots, 2nd ed.

在静态环境，且栅格地图已知的情况下，用以下哪些算法可得到最短路径

- ☐ A 广度优先
- ☐ B 深度优先
- ☐ C Dijkstra算法
- ☐ D A*算法

都是静态图搜索方法

提交

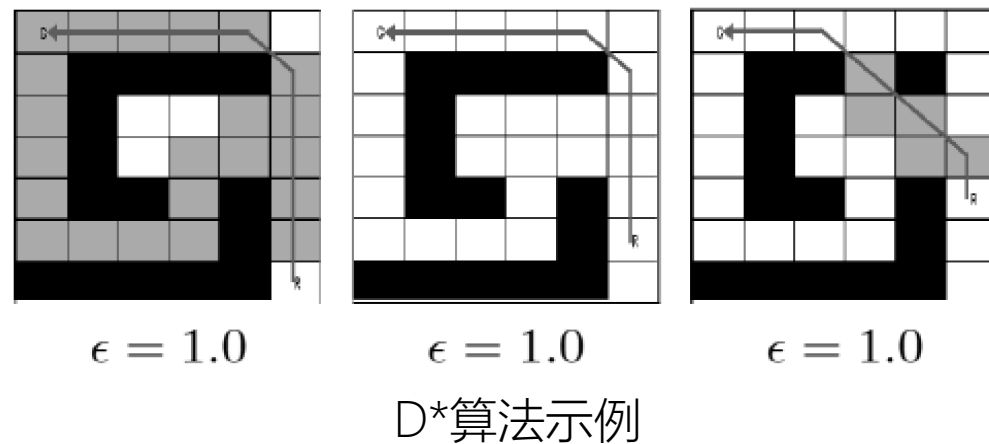
2.1.2 图的搜索

- D*搜索（动态图搜索算法）

在初始状态已知粗略环境地图。

1. 用Dijkstra算法从目标节点G向起始节点搜索找到最优路径，同时储存目标点G到各个节点的最短路径和各节点的启发代价 $h(n)$

2. 机器人沿最优路径移动，当观察到环境变化时，更新相关节点的启发代价 $h(n)$ ，求新的最优路径（仅需要考虑地图变化带来的影响）：用D算法从目标节点（或变化节点的上一层节点）开始扩展，一直到机器人的当前状态。



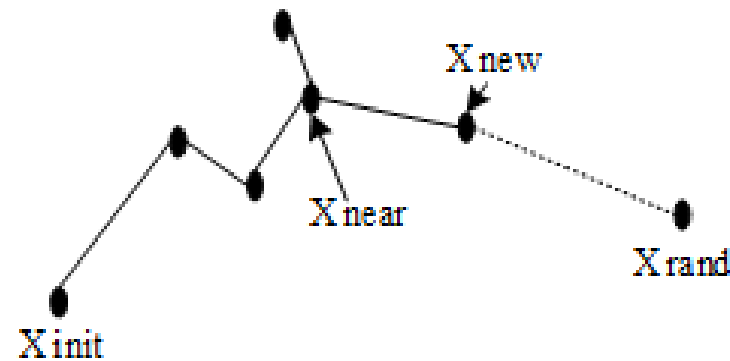
*适用于动态环境中的路径规划，可用于避障，是火星探测器采用的寻路算法

2.1.2 图的搜索

- 快速搜索随机树 (Rapidly-exploring Random Trees)

已知起始点 X_{init}

树的生成： 在自由空间中随机撒一个点，记作 X_{rand} ；连接 X_{init} 和 X_{rand} ，若它们的连线不经过障碍区域，则沿该连线从 X_{init} 向 X_{rand} 移动一定距离，得到新节点 X_{new} ，连接 X_{init} 和 X_{new} ，从而得到一棵最简单的树。



RRT生成示意图

树的扩展： 在自由空间中随机撒一个点，仍记作 X_{rand} ，在已经存在的树上找一个离 X_{rand} 最近的节点 X_{near} ，如果 X_{near} 和 X_{rand} 的连线没有障碍，则沿着这条线，从 X_{near} 向 X_{rand} 移动一定的距离，得到新的节点 X_{new} ，该节点被添加到已经存在的树上。重复上述过程，直到目标点（或其附近的点）被添加到树上，

规划： 用某种搜索算法在树上找到一条从起始点到目标点的路径。

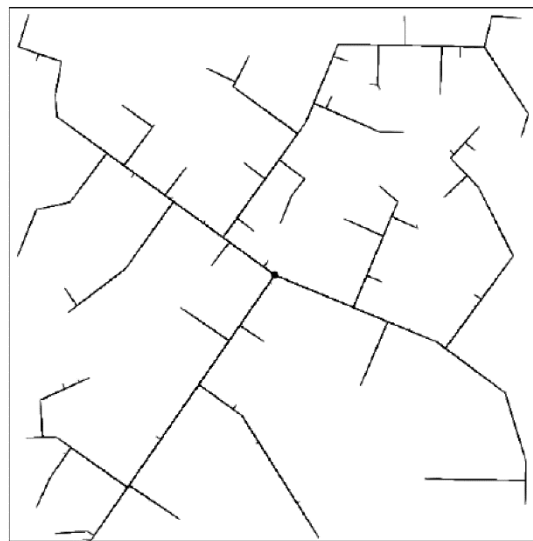
快速搜索随机树方法是一种随机图搜索方法，一般情况下，用该方法得到的是最短路径吗？

- ☐ A 是
- ☒ B 否

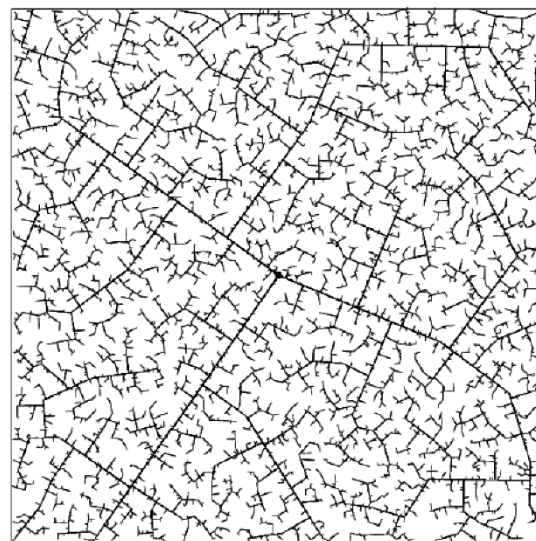
提交

2.1.2 图的搜索

- 快速搜索随机树 (Rapidly-exploring Random Trees)



45 iterations



2345 iterations

RRT演化过程

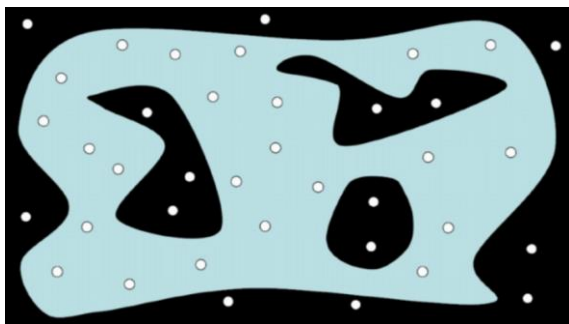
RRT: 适用于搜索非凸的高维度的空间，甚至可用于解决微分约束条件下的路径规划或运动规划

2.1 图搜索方法

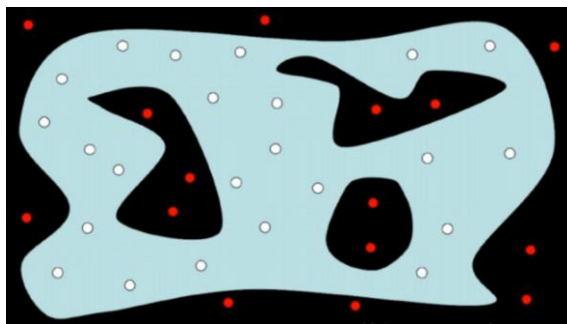


2.1.2 图的搜索

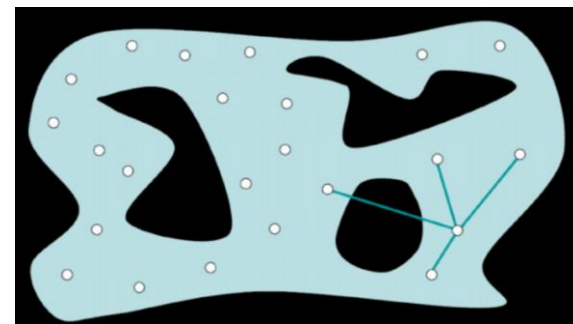
- 随机路线图法 (Probabilistic Roadmaps)



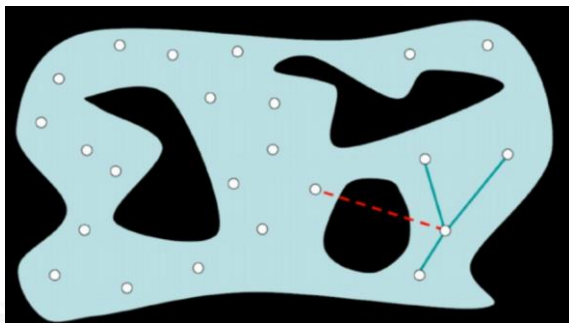
(1) 随机采样



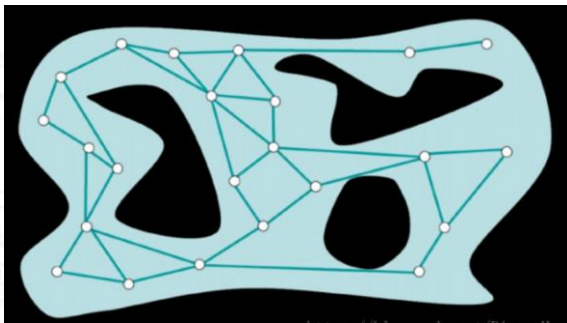
(2) 去掉障碍区域的样本



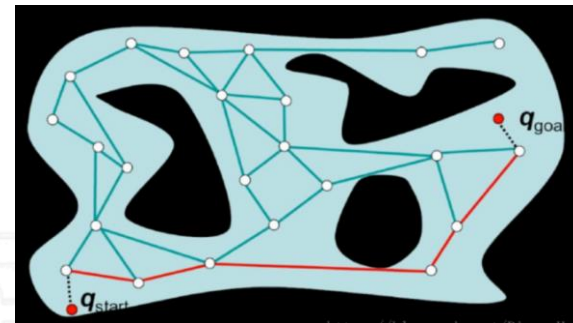
(3) 将每个样本与k个最近邻相连



(4) 去掉穿越障碍区域的连接



(5) 得到一幅随机路线图 (PRM) (无向图)



(6) 将起点和终点与PRM相连
(7) 用A*算法搜索最优路径

基于 随机采样技术 的 PRM法 可以有效解决 “高维空间” 和 “复杂约束” 中的路径规划问题。

2.2 势场规划方法



在地图上人为建立一个引导机器人到达目标位置的场

- 将机器人看作人工势场中的一个点（可不考虑朝向）。
- 目标对机器人有引力，障碍物对机器人有斥力。
- 施加于机器人的合力平滑地引导机器人趋向目标，同时避免碰撞已知的障碍物。

注意：人工势场法产生的不仅是路径规划，还是机器人的运动控制律。

2.2 势场规划方法



基本思想： 机器人被吸引向目标，同时被已知的障碍物所排斥

记机器人的位置为 (x, y) , 设势函数 $U(q)=U_{att}(q)+U_{rep}(q)$ 可微, 则作用于位置 $q=(x, y)$ 的力为:

$$F(q)=-\nabla U(q)=-\begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{bmatrix}$$

这里 $\nabla U(q)$ 表示在点 q 处 U 的梯度向量。

2.2 势场规划方法



例: $U_{att}(q) = \frac{1}{2}k_{att} \cdot \rho_{goal}^2(q)$ where $\rho_{goal}(q) = \|\rho - \rho_{goal}\|$ 则有 $F_{att}(q) = -\nabla U_{att}(q)$

$$= -k_{att} \cdot \rho_{goal}(q) \nabla \rho_{goal}(q)$$
$$= -k_{att} \cdot (q - q_{goal})$$

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right)^2 & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases}$$

$\rho(q)$ 是 q 点到障碍物的最短距离



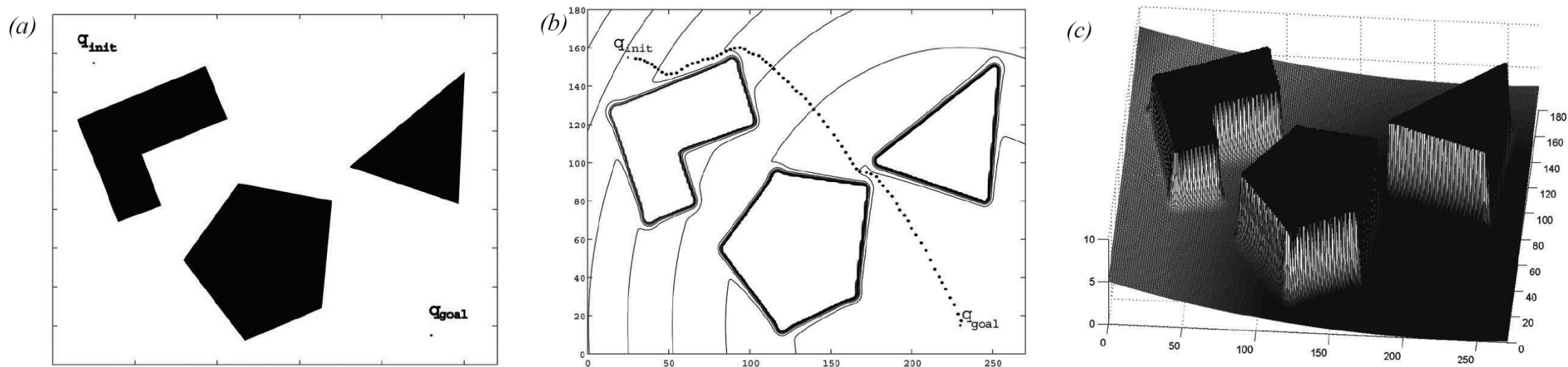
如果障碍物的边界是凸的, 且分段可微

$$F_{rep}(q) = -\nabla U_{rep}(q)$$

$$= \begin{cases} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \frac{q - q_{obstacle}}{\rho(q)} & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases}$$

在合力 $F(q) = F_{att}(q) + F_{rep}(q)$ 的作用下, 机器人远离障碍物并向目标移动
理想情况下, 通过设置一个正比于场力向量的机器人速度向量, 可平滑地引导机器人趋向目标。

2.2 势场规划方法

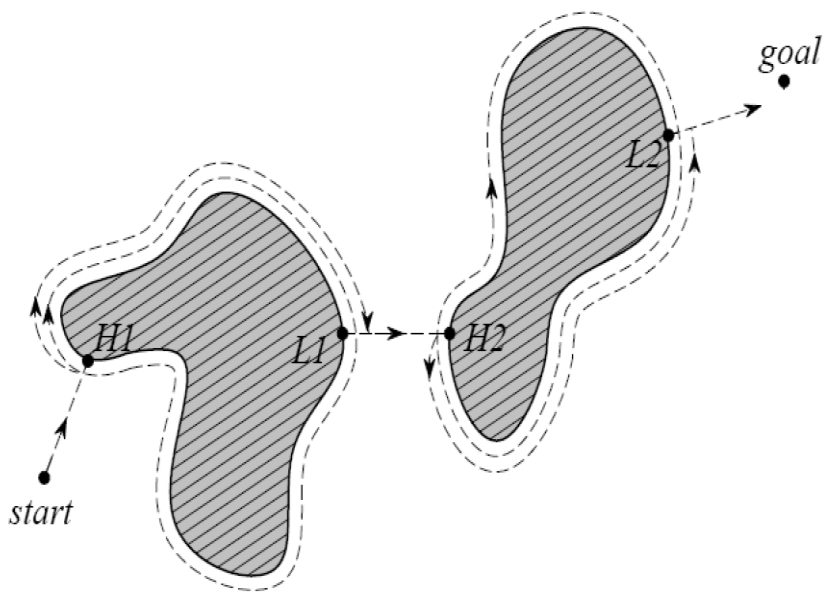


人工势场示例图 (a)环境构型； (b)等势图和由势场产生的路径； (c)势场大小示意

局限性：可能出现局部极小；如果障碍物是凹的，还可能产生振荡。

3.1 Bug算法

基本思想： 在机器人向目标运动的过程中，跟踪各障碍物的轮廓，从而绕开它。



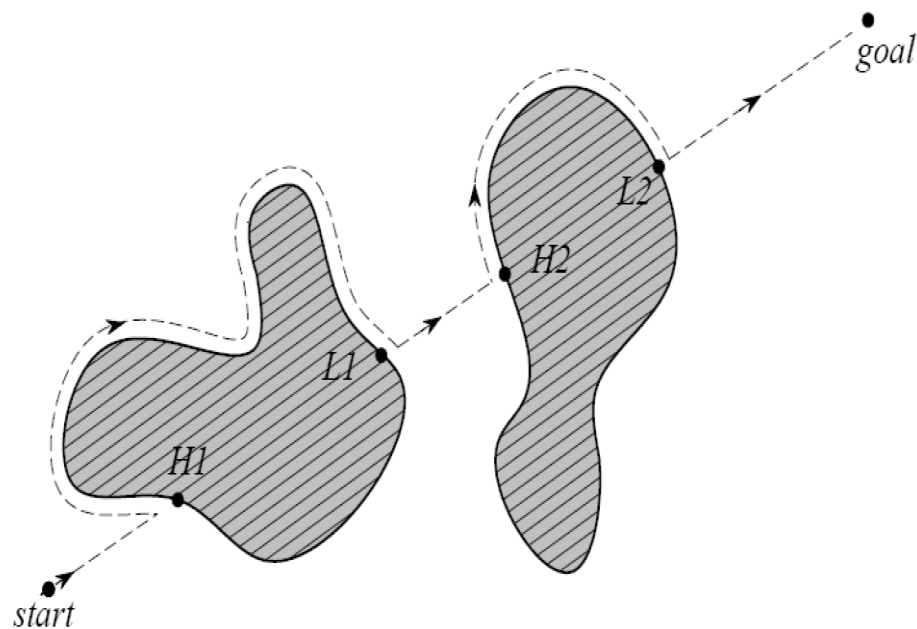
Bug1 算法示意图

Bug1：最短距离原则到达障碍，跟踪障碍物轮廓运动，并从距离目标最短距离点离开

效率很低，但可保证机器人可到达任意可达的目标。

3.1 Bug算法

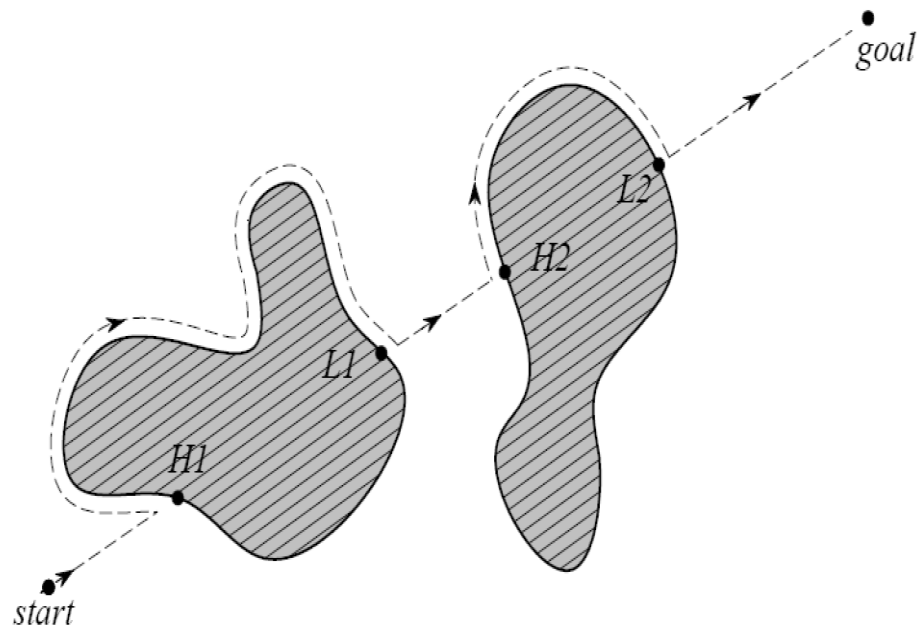
Bug2：指向目标点原则，遇到障碍则跟踪障碍物轮廓，直到能再次沿着指向目标点的方向运动。



Bug2 算法示意图

3.1 Bug算法

Bug2：指向目标点原则，遇到障碍则跟踪障碍物轮廓，直到能再次沿着指向目标点的方向运动。



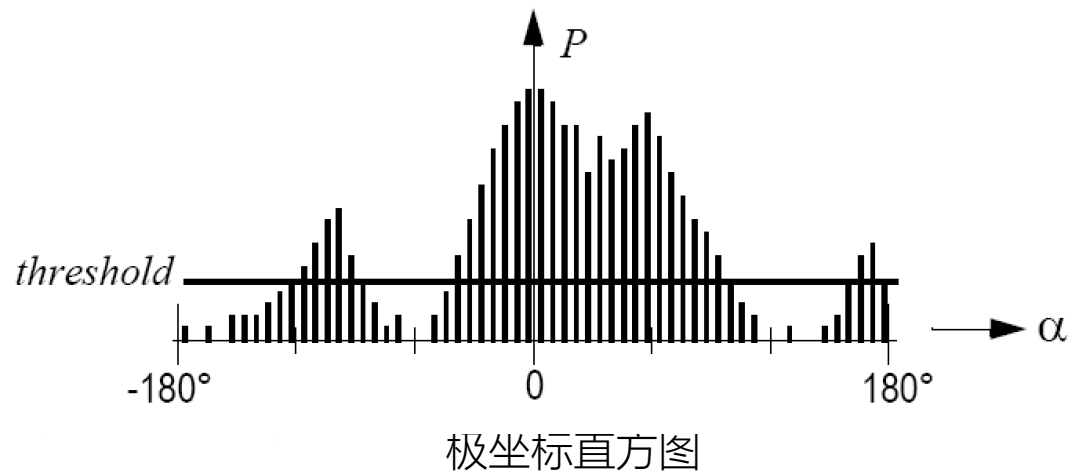
Bug2 算法示意图

比较而言，Bug1执行穷举搜索以找到最佳离开点或接入点，更保守；Bug2使用机会主义方法。当障碍物很简单时，Bug2的贪婪方法会带来很快的回报，但是当障碍物很复杂时，Bug1通常会产生更好的性能

3.2 向量场直方图(vector field histogram, VFH)

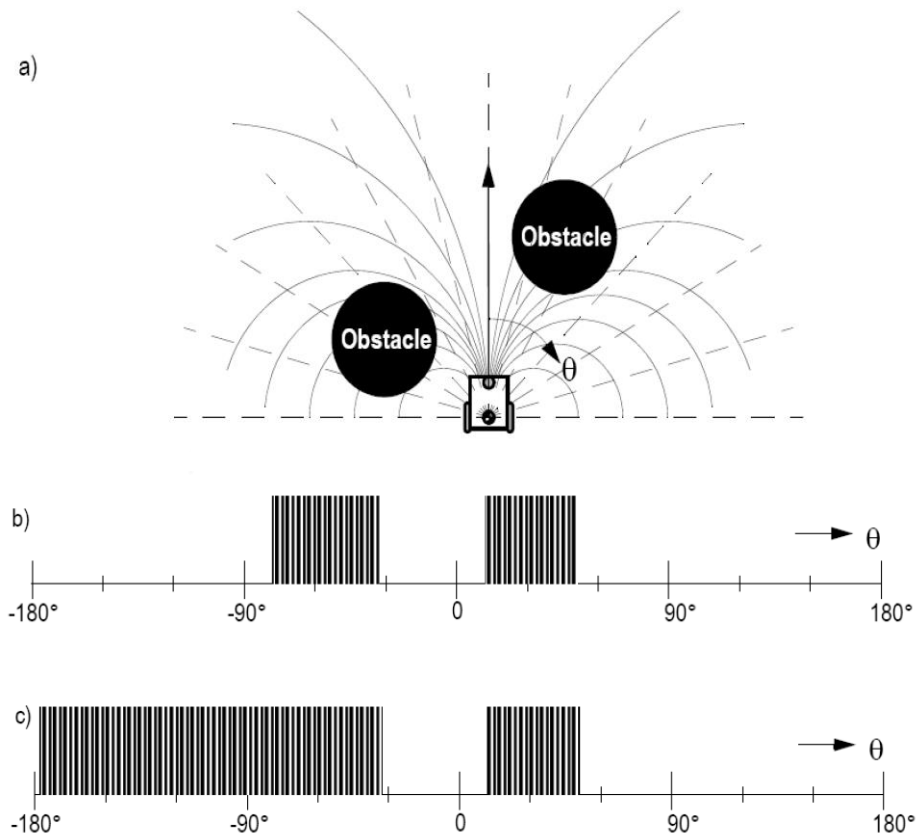
- 基于传感器数据构建机器人周围的局部栅格地图，并产生一个极坐标直方图，
- 确定行驶方向。首先确定车辆可以通过的所有通路，然后对每一个候选通路计算代价函数，选择代价最低的通路。例如：

$G = a \cdot \text{目标方向} + b \cdot \text{轮子方向} + c \cdot \text{以前方向}$
目标方向 = 与目标方向的偏差；
轮子方向 = 新方向和当前轮子方向之差；
以前方向 = 以前所选方向和新方向之差。



3.2 向量场直方图(vector field histogram, VFH)

VFH+: 考虑机器人运动学的限制, 假设将机器人将沿圆弧或直线运动。因此, 障碍物阻挡了穿过该障碍物的所有轨迹。所有运动学上被阻塞的轨迹 (c) 都被适当地予以考虑。



VFH+ 极坐标直方图举例: (a)机器人与障碍物;
(b)极坐标直方图; (c)被屏蔽的极坐标直方图

路径规划

- 图搜索路径规划方法

- 1. 图的构建:

- ① 路线图法: 可视图、沃罗诺伊图
 - ② 单元分解法: 栅格地图

- 2. 图的搜索:

- ① 广度优先、深度优先、D算法、A*算法 (静态环境)
 - ② D*算法 (动态环境)
 - ③ RRT、PRM (复杂环境)

- 势场路径规划方法

避障

- Bug

- 1. Bug1

- 2. Bug2

- VFH



清华大学 自动化系
Department of Automation, Tsinghua University

End of 智能机器人运动规划